Faculty of Engineering, Computer and Mathematical Sciences
SCHOOL OF MECHANICAL ENGINEERING

Learning Acoustics and the Boundary Element Method Using Helm3D and GiD

TUTORIAL MATERIAL

November 28, 2005

Laura A Brooks[†] and Richard C Morgans[††]

[†]email: laura.brooks@mecheng.adelaide.edu.au
[††]email: rick.morgans@gmail.com

source: "Piled Higher and Deeper" by Jorge Cham - www.phdcomics.com

Learning Acoustics and the Boundary Element Method Using Helm3D and GiD

Brooks, L A & Morgans, R C November 2005
Active Noise and Vibration Control Group
School of Mechanical Engineering
The University of Adelaide
SA 5005 Australia

Typeset by the authors with the LaTeX 2_ε document preparation system. Please submit and errors, suggestions or modifications to the authors. Printed in Australia.

Précis

The Boundary Element Method (BEM) is a powerful tool which has become an important and useful numerical technique applied to problems in acoustics. It is particularly useful for analysing sound radiation and acoustic scattering problems. Numerous commercial BEM codes with graphical user interfaces (GUIs) and mesh generators exist; however these are relatively expensive, which discourages their use by academic institutions and smaller companies. Helm3D is a three-dimensional BEM code available with purchase of a relatively inexpensive book, but the command file driven interface is difficult to learn and some mechanism to generate the mesh is required. In addition, there is a limited availability of suitable tutorial material, so the uptake of BEM throughout the acoustics community has so far been limited.

A GUI interface to a low cost commercial mesh generator (GiD) has been developed for the Helm3D code. This tutorial material guides the user through the use of the GUI to solve BEM problems. Step-by-step instructions which explain how to input each model, apply boundary conditions and postprocess the results are given. Comparisons with analytical solutions are given when possible.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The acoustic Boundary Element Method (BEM) has been used to solve a wide range of practical problems in acoustics, such as the modelling of sound generated by loudspeakers [1] and [2] or received by microphones [3], the sound power radiated by a particular structure such as an engine valve cover [4] or a fan [5], and the sound scattered by hard structures [6]. Numerous commercial codes that implement acoustic BEM exist; however the licensing costs are prohibitively expensive for casual users, limiting the uptake of this technology by the wider acoustics community. There exist numerous non-commercial acoustic BEM codes, such as those associated with the book edited by Wu [7]. These source codes exist as pedagogical examples for teaching the basics of BEM at an advanced undergraduate or postgraduate level. They are written in Fortran 77 and are available the CD accompanying the book. They are fully featured and capable of solving practical problems [8]. These non-commercial codes, whilst readily available with the purchase of the book, have not gained widespread use for a number of reasons: the interface has traditionally been command file driven and requires access to some form of pre and postprocessor, and there is a limited availability of suitable tutorial material. Thus it was realised that there was a need for:

- an easy to use, freely available interface to an acoustic BEM code, and
- a well written, step by step tutorial on the use of BEM to solve simple relevant acoustic problems.

A GUI interface to Helm3D within the GiD environment has been developed, meeting the first requirement. This tutorial satisfies the second requirement, presenting step-by-step instructions that teach the user funda-

mental acoustic concepts, BEM concepts and how to use the GUI interface to solve BEM problems.

1.1 GiD

GiD is a general-purpose, fully featured finite element pre and post processor developed over a number of years by the International Centre for Numerical Methods in Engineering (CIMNE) in Barcelons, Spain. It has extensive geometry creation features as well as CAD import (IGES and others), supports the meshing of many different element types, the application of boundary conditions and has a postprocessing capability for viewing results.

The academic version of this program is freely downloadable, the only restriction being limited to 700 3D elements. Fortunately for BEM, this is a reasonable size and many useful acoustic problems can be solved.

It is highly recommended that those who are unfamiliar with GiD should download the GiD user manual from the website (http://gid.cimne.upc.es).

1.2 How to install Helm3D and GiD

Both the pre and postprocessor GiD and the BEM code Helm3D are required. Although GiD can be operated using Linux, Windows is currently the only platform supported. Please contact the authors if Linux compatibility is required.

The book *Boundary Element Acoustics: Fundamentals and Computer Codes* [7], including a CD containing a PC executable of Helm3d (helm3d.exe), as well as F77 source code, is available from WITPress. (http://www.witpress.com/acatalog/5709.html)

The program GiD can be downloaded from its homepage. (http://gid.cimne.upc.es). The program will be downloaded as an executable. To unpack and install GiD on your computer simply run the executable file and follow the step by step instructions of the setup procedure.

Helm3D.zip can be downloaded from the University of Adelaide Active Noise and Vibration Control (ANVC) Group homepage. (http://www.mecheng.adelaide.edu.au/anvc/publications.php). Once you have downloaded GiD you need to unzip Helm3D.zip into the GiD 'problem types' folder (GiD\Gid7.2\problemtypes). A folder entitled helm3d.gid containing the contents of the zip file should be created. The helm3D executable (helm3d.exe) obtained from the CD accompanying the aforemen-

tioned BEM book also needs to be copied and placed within the helm3d.gid folder.

1.3 Tutorial structure

The tutorial guides the user through BEM modelling with eight problems, each introducing different aspects of:

- fundamental concepts in acoustics,
- BEM specific concepts, and
- using the GiD-Helm3d interface.

The tutorial material comprises step-by-step instructions which explain how to input each model, apply boundary conditions and postprocess the results. Comparisons with analytical solutions are given when possible. By the end of the tutorial, the user should have had an introduction to these fundamental concepts in acoustics:

- one-dimensional standing waves,
- one-dimensional travelling waves,
- impedance (sound absorbing) boundary conditions,
- modes in a rectangular room,
- modes in more complex spaces,
- one-dimensional spherical waves,
- sound radiation from a sphere, and
- sound radiation from more complex shapes.

The user should understand these BEM specific concepts:

- advantages and disadvantages when compared to other techniques,
- interior versus exterior problems,
- element types,
- mesh size (6 elements per wavelength),

- non-uniqueness difficulty (CHIEF points),
- symmetry, and
- direction of normals.

The user should also have a working knowledge of these GiD-Helm3d interface concepts:

- inputting the geometry into GiD directly,
- importing CAD data into GiD for meshing,
- flipping surface normals,
- meshing the geometry,
- applying boundary conditions,
- solving the problem through the GiD interface to Helm3d, and
- post-processing results through GiD.

Figure 1.1 shows the breakdown of the tutorials. Two application areas are addressed: interior acoustics and external acoustic radiation. Simple problems with analytical solutions are introduced. The power of BEM is then demonstrated through application to more realistic problems. Step-by-step instructions on how to solve each of the eight tutorial problems are given in the subsequent chapters.

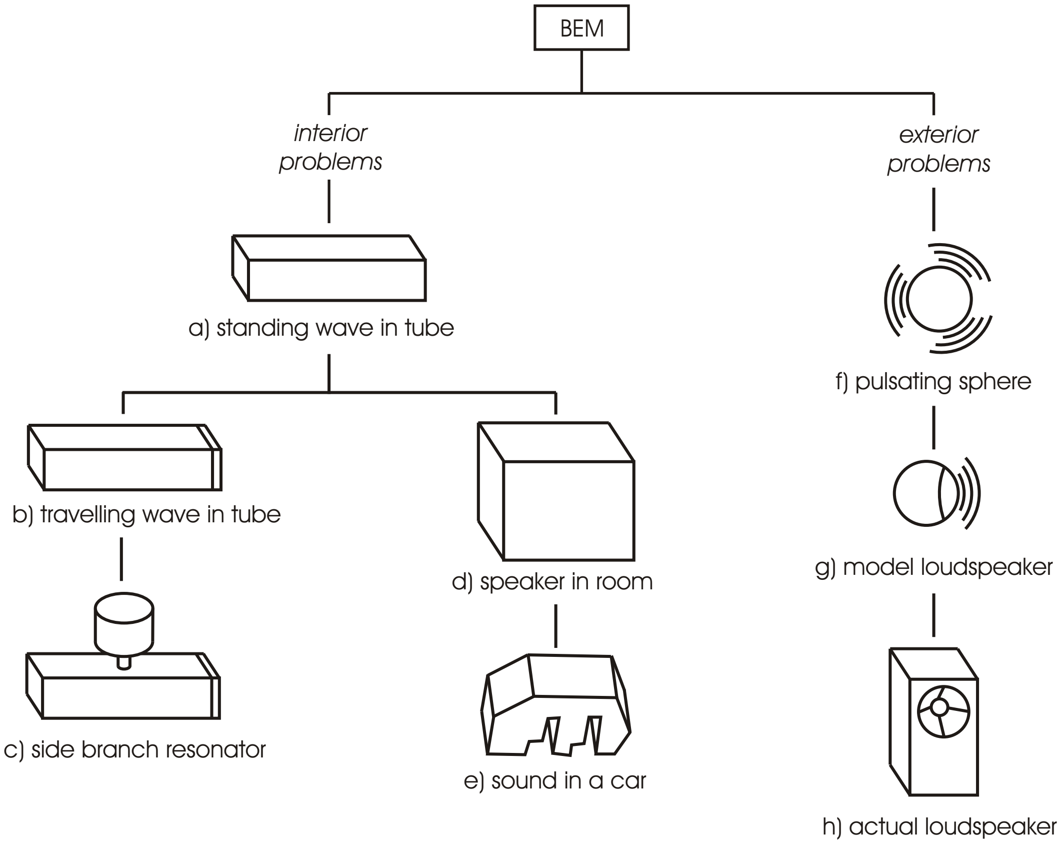

Figure 1.1: Breakdown of the tutorial problems.

1.4 Nomenclature

Within this tutorial, instructions written in bold text indicate that the user should select from the main title bar at the top of the screen. For example, **Meshing > Generate** is asking the user to select 'Meshing' from the title bar using the mouse and then to scroll down to 'Generate' and then select this. Commands enclosed in triangle brackets are to be typed on the

keyboard. For example <:> is asking the user to type a colon. Commands to hit or press 'enter' or 'esc' are asking the user to hit the enter or escape key on the keyboard respectively. If the user is asked to select something enclosed in quotations, then they need to click on the appropriate button using the mouse. For example, select 'OK' is asking the user to use the mouse to click on the 'OK' button on the screen.

Chapter 2

Example One: Standing Wave in a Duct

The first example (see Figure 1.1.a) is a simple model of a 1D standing wave in a rigid walled duct. A 1D standing wave is created when a source of constant velocity is operated at one end of a closed tube. The sound emitted from the source experiences multiple reflections from each end of the tube. The resulting forward and back propagating waves combine to form a 'standing wave' of high amplitude.

This problem introduces the very simple geometry of a long rectangle. One of the dimensions is much larger than the other two, enabling the assumption of 1D plane propagating waves to be valid, which simplifies the theoretical analysis at low frequencies (below the cut-on frequency of higher order modes). Velocity boundary conditions, the required direction of normals and meshing are introduced. How the accuracy of results can be affected by mesh resolution is also demonstrated. Results obtained from the numerical model are then compared to the analytical solution.

Open up a new project (**Files > New**) and save it (**Files > Save**) in your working directory as 'stand.gid'. This will create a folder in which all of the files generated using GiD will automatically be saved.

The first step is to construct the rectangular tube depicted in Figure 2.1.

Open up an auxiliary window from which coordinates can be easily entered (**Utilities > Graphical > Coordinates Window**).

Chose the option of creating a point (**Geometry > Create > Point**).

Enter the eight points in Table 2.1 by typing their coordinates in the coordinate window.

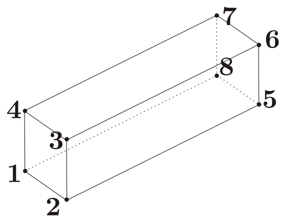

Figure 2.1: Rectangular tube.

Table 2.1: Coordinates of the duct vertices.

point	coordinates
1	(0,0,0)
2	(1,0,0)
3	(1,1,0)
4	(0,1,0)
5	(1,0,10)
6	(1,1,10)
7	(0,1,10)
8	(0,0,10)

Click 'apply' after entering the coordinates of each point (see Figure 2.2). Click 'close' once all points have been entered.

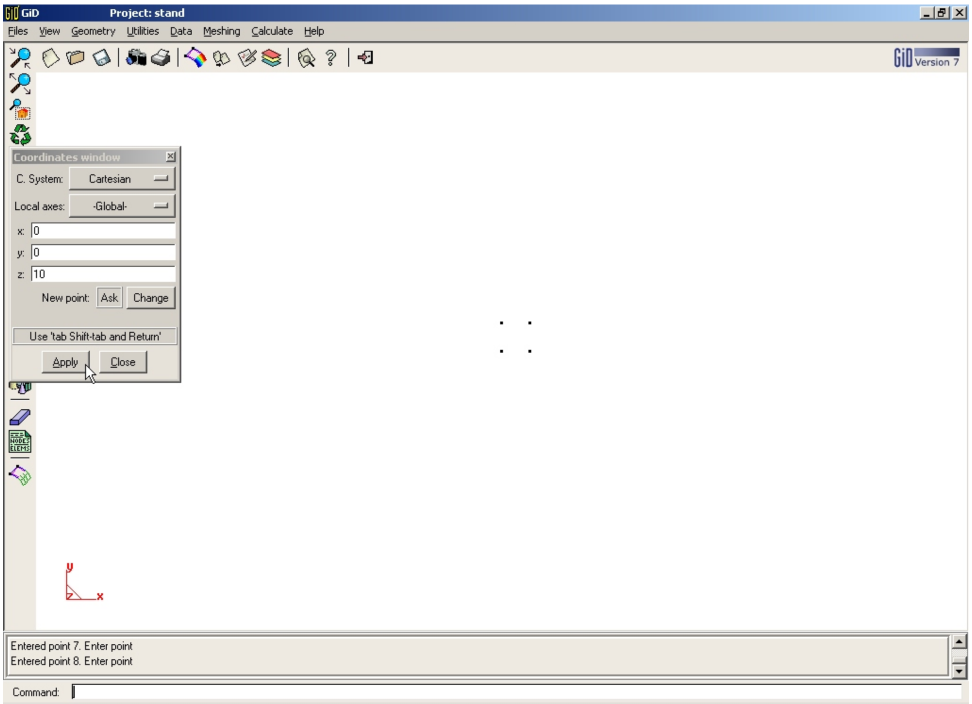

Figure 2.2: Entering the point coordinates.

Using the trackball (**View > Rotate > Trackball**), rotate the coordinate system until all eight points can clearly be seen in a 3D view.

Centre the image and ensure that the zoom is reasonable for the window size (**View > Zoom > Frame**).

Construct a rectangular prism by joining the nodes as depicted in Figure 2.3. To do this, select the option to create a line (**Geometry > Create Line**). Right click the mouse and select **Contextual > Join C-a** (or hit 'ctrl-a'). Click on the points to join the lines. To start a line from a point different to the finish point of the last line, right click the mouse and select **Contextual > Escape** (or hit the 'esc' key) before clicking the first point of the new line.

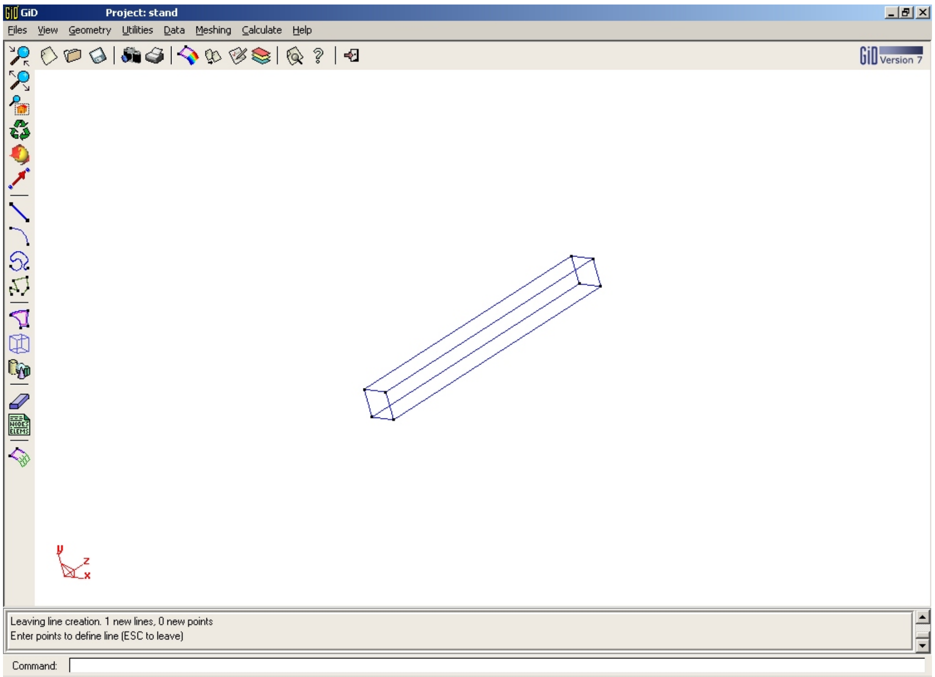

Figure 2.3: Joining the points to form a prism.

Now that the basic geometry has been defined, the surfaces of the boundary problem need to be defined. To do this select **Geometry > Create > NURBS surface > By contour**. Click on the four lines defining the edges of one of the prism surfaces. Upon selection they will be highlighted in red. If an incorrect line is accidently selected it can be unselected by clicking on it once more. Once the four lines are highlighted hit 'esc'. a purple rectangle will appear inside of the original boundary, indicating the existence of a surface. Use the same procedure to define each of the other five prism surfaces as depicted in Figure 2.4.

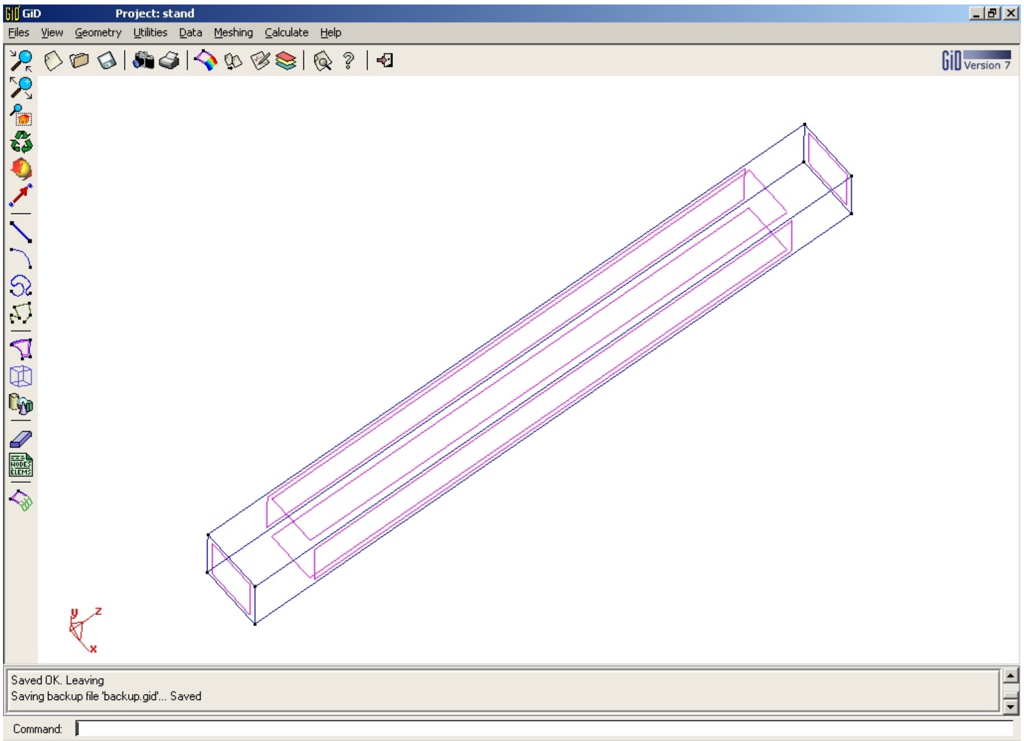

Figure 2.4: Constructing the prism surfaces.

The next step is to check the surface normals. Select **Utilities > Draw Normals > Surfaces**. To select all surfaces type <:> in the command line and then press 'enter' (or select the entire model by clicking and dragging the mouse from one corner to the diagonally opposite corner of the screen). Some of the surfaces may point into the prism whilst others may be pointing out (see Figure 2.5).

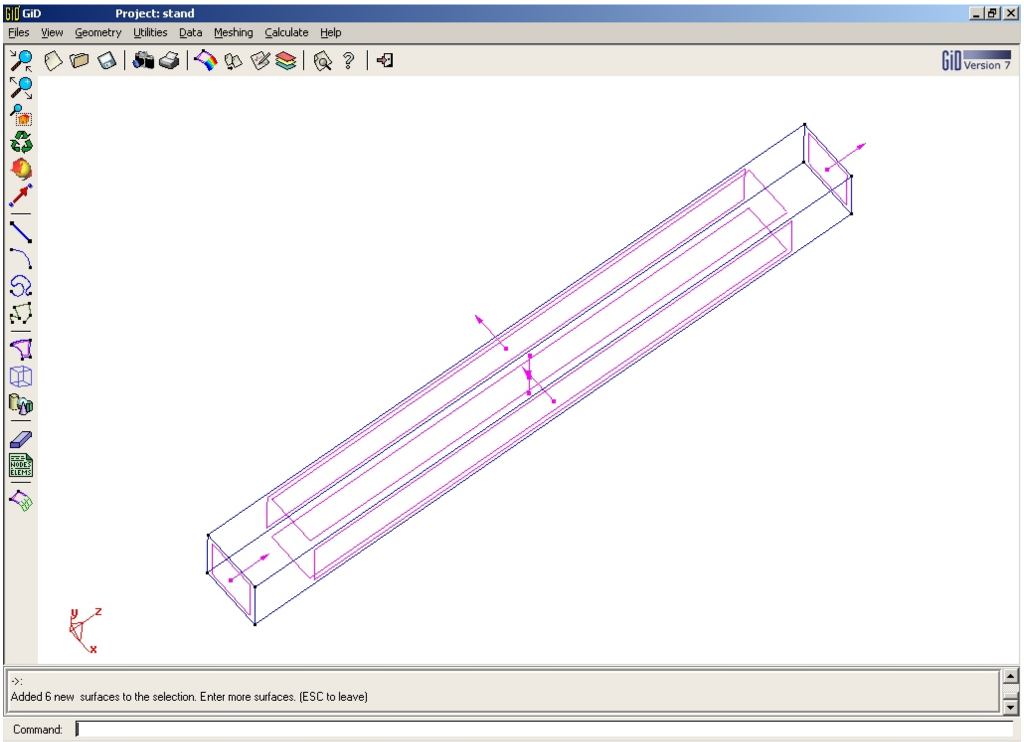

Figure 2.5: 'Draw Normals' environment.

For an internal boundary element problem all surfaces must face out. Whilst still in the 'Draw Normals' mode, right click the mouse and select **Contextual > Swap some**. Click on all of the surfaces that have an inward pointing normal until all surfaces are oriented in the correct direction (see Figure 2.6). Hit 'esc' to leave the 'Draw Normals' mode.

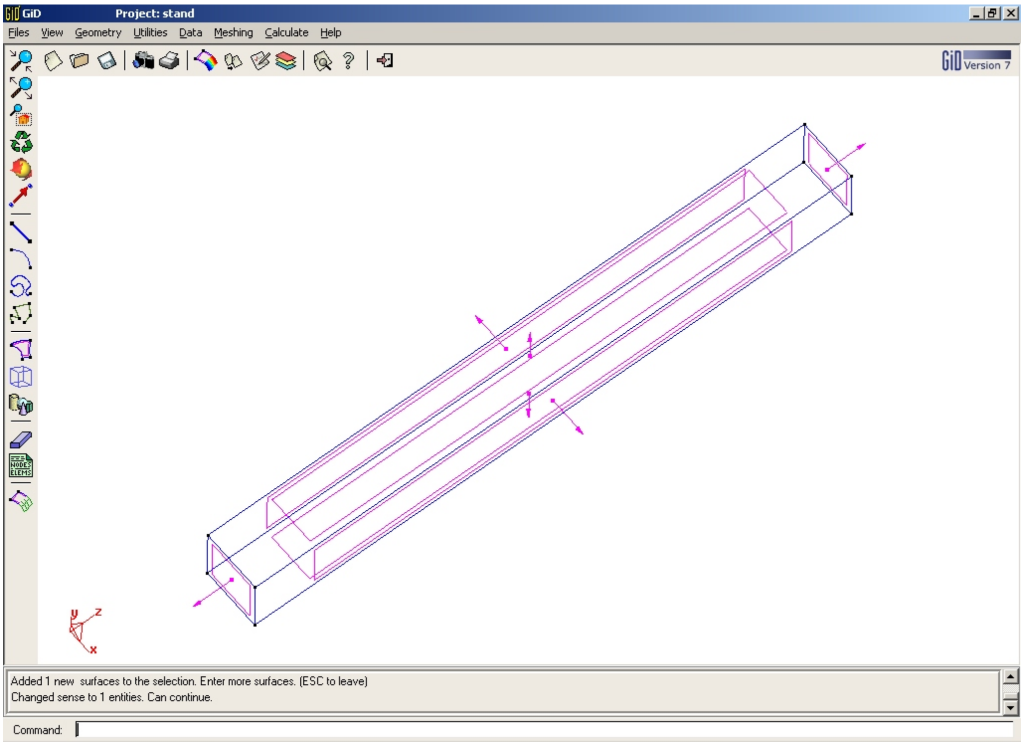

Figure 2.6: All normals are oriented correctly for an internal BEM problem.

The next step is to define the problem as a Helm3D BEM problem. To do this select (**Data > Problem type > helm3d**). A dialog window warning the user that 'all data information will be lost' will appear. Select 'OK'. Once the problem type has been chosen, the boundary conditions need to be defined. The only boundary conditions required for the standing wave (ie. magnitude 1, phase ϕ) problem are velocity boundary conditions. A velocity of 1+0i is needed to mimic the piston at the entrance of the tube. All other surfaces need to be set to zero velocity. To do this select (**Data > Conditions**). Click on 'Velocity'. Set both the real and imaginary velocity components to zero. Click on 'assign'. Type <:> in the command line to select all surfaces and select 'finish'. Change the real velocity component to 1. Click on assign and select the surface at z=0 (as depicted in Figure 2.7). Select 'finish'. The boundary conditions have now been set.

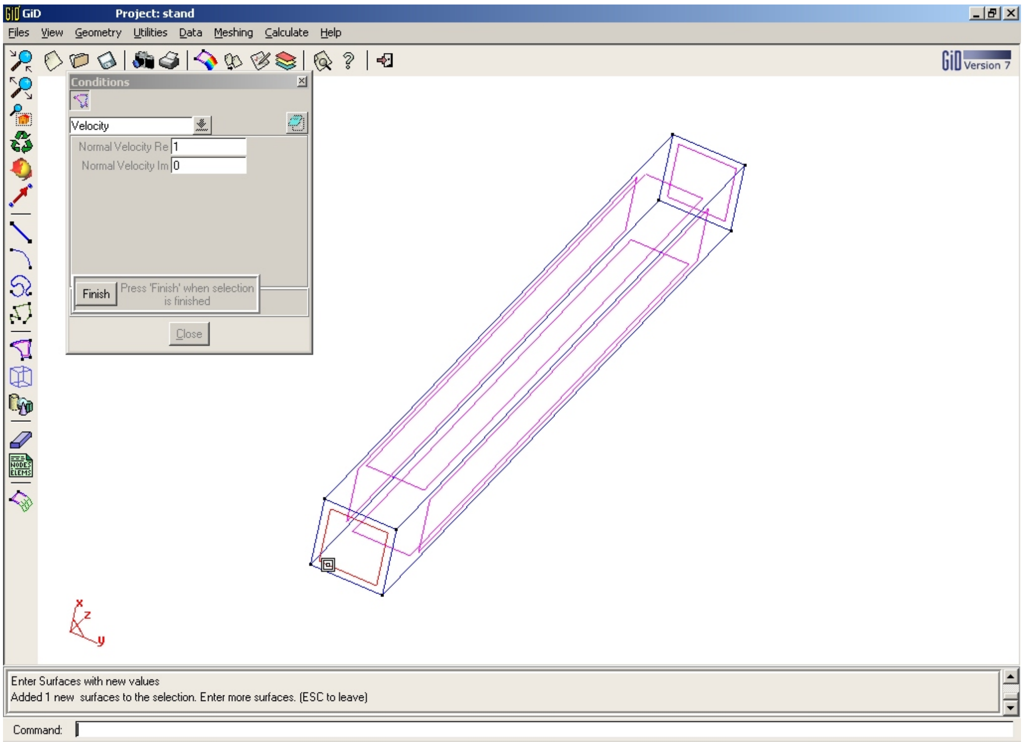

Figure 2.7: Setting the z=0 velocity boundary condition.

Select (**Data > Problem data**). Give the project a title of <stand>. Ensure the boundary type is selected to be 'internal' and that the density and speed of sound are selected to be 1.21 and 343 respectively. Set the 'frequency start' and 'frequency stop' to be 25.725, corresponding to the second resonance frequency of the tube and then click 'Accept data' and 'Close' (see Figure 2.8).

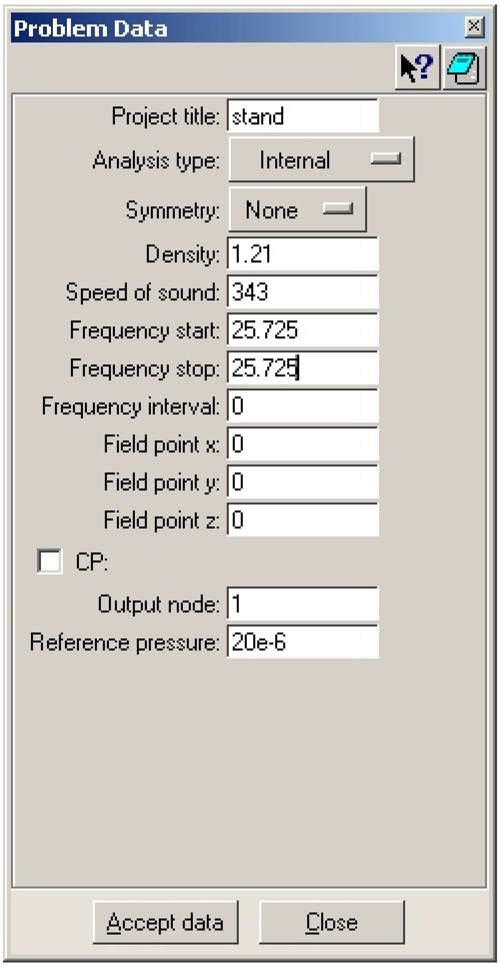

Figure 2.8: Defining the problem data.

The next step is to mesh the boundary of the tube. Select (**Meshing > Generate**). A dialog box will appear asking you to 'Enter the size of elements to be generated'. Type in <2>. A dialog box will appear which states that 52 triangle elements have been created. Press 'OK' and the mesh will appear (Figure 2.9).

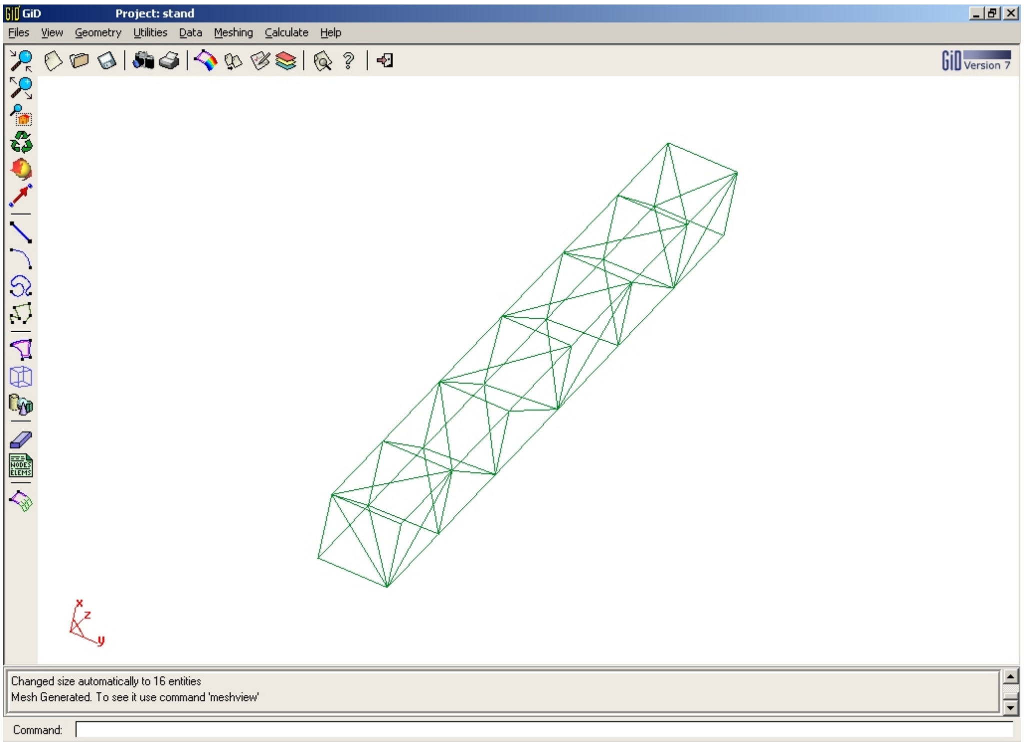

Figure 2.9: View of the meshed tube.

A solution to the problem can now be generated by selecting (**Calculate > Calculate**). A dialog box will appear telling you once the solution is done (Figure 2.10). Click 'OK'.

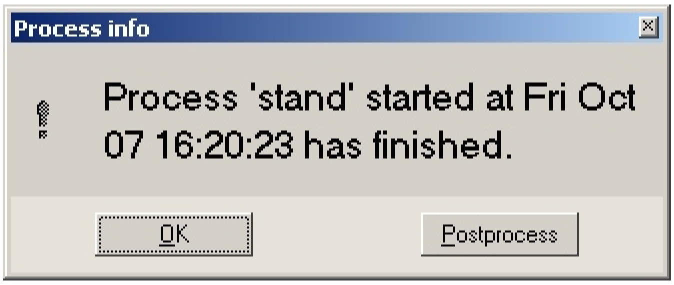

Figure 2.10: Process information dialog box.

To review the results, once must enter the postprocessor. To do this select (**Files > Postprocess**). To display the pressure over the boundary of the duct select (**View results > Contour fill > Press amp**) (Figure 2.11).

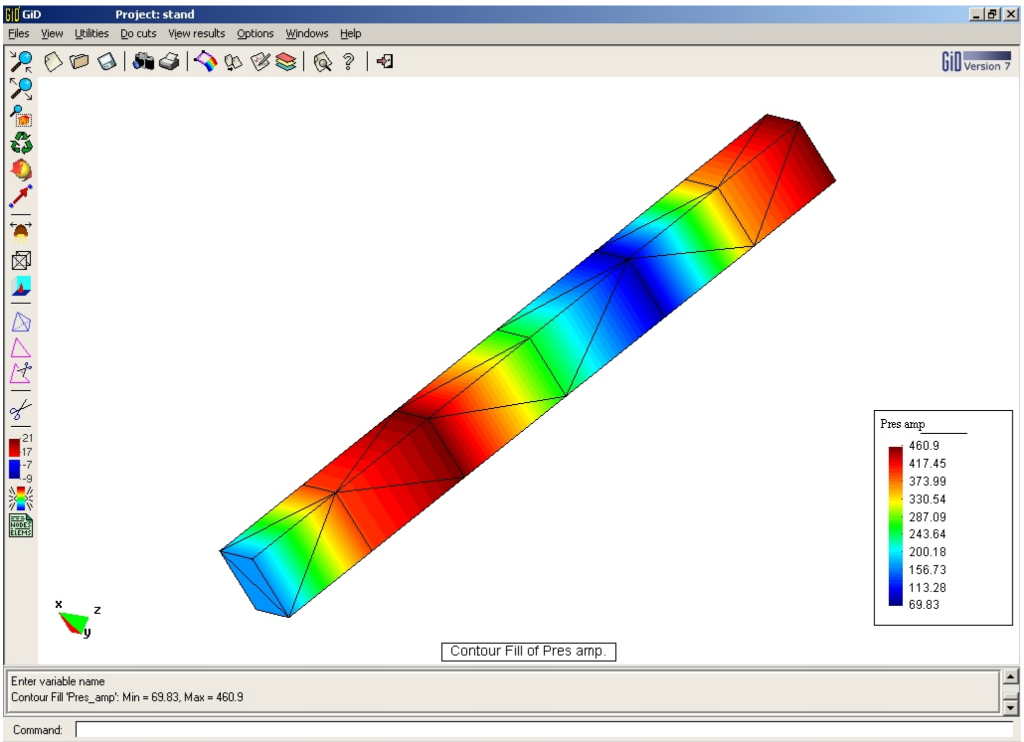

Figure 2.11: Pressure amplitude over boundary of duct.

As expected, a standing wave corresponding to the second resonance frequency of the tube is observed. To see how the real and imaginary pressure components vary along the tube or to see the total sound pressure level in dB, select (**View results > Contour fill**) and chose the desired parameter.

Although the BEM problem was solved only at one frequency, the solution can be swept over a frequency range in order to see the frequency dependance of the solution. To do this, you first need to return to the preprocessor (**Files > Preprocess**). The frequency range can then be changed within the problem data environment (**Data > Problem data**). Change 'frequency start' to 1, 'frequency stop' to '110' and 'frequency interval' to a suitably such as 1 as depicted in Figure 2.12 (the smaller the increment, the higher the resolution of the result, but the necessary computational effort also increases).

If the problem data or conditions are changed the model must always be remeshed before the new solution is obtained. To do this select (**Meshing > Generate**). A warning will appear alerting you that the old mesh will be erased and asking you whether to continue with the mesh. Click 'OK' for this and the following two dialog boxes.

Generate a new solution to the problem (**Calculate > Calculate**). A greater period of time will elapse before the dialog box telling you that the solution is done appears. This is due to a separate solution having to be generated at each frequency increment.

Once the solution is complete you can once again enter the postprocessor to review your results (**Files > Postprocess**). The pressure at the field point (0,0,0), ie. the point of excitation at each frequency, can be viewed by opening the 'output.fdat' file (using any simple text editor) which will have appeared in your working folder (the folder in which you have saved 'stand'). The first line contains the headings of each column of data. Of greatest interest in this case are the first column which is the frequency and the ninth column which is the pressure amplitude (in Pascals) of the field point (in our case the point of excitation). As the boundary condition at the point of excitation necessitates unit velocity amplitude at this location, the specific acoustic impedance, which is the ratio between the acoustic pressure and the particle velocity is simply the magnitude of the pressure at this point.

The theoretical resonance frequencies of the system are simply the resonances of an open-closed duct and are given by:

$$f_n = \frac{nc}{2l} \tag{2.1}$$

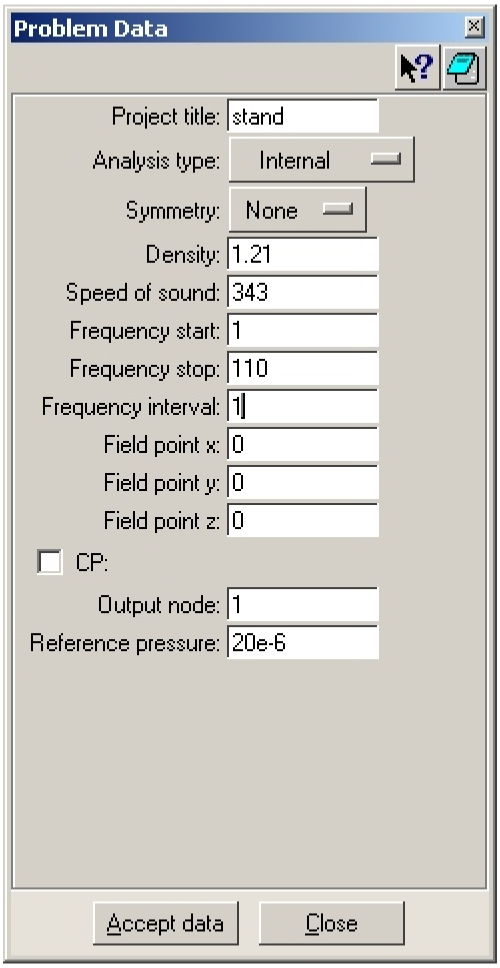

Figure 2.12: Altering the problem data to sweep over a frequency range.

where n is the mode number, c is the speed of sound and l is the length of the duct. The analytical specific acoustic impedance at the excitation location is:

$$Z_s = 0 - i\rho c \cot(kl) = \frac{p}{v} \tag{2.2}$$

where $i = \sqrt{-1}$, ρ is the density of the medium, k is the wavenumber, p acoustic pressure and v is the particle velocity. By solving the BEM problem over a range of frequencies as you have done, the theoretical specific acoustic impedance and the BEM specific acoustic impedance (the ratio between the acoustic pressure and the particle velocity) at the point of excitation can be obtained. An example comparing the theory and BEM solutions was

constructed using Matlab (see Figure 2.13).

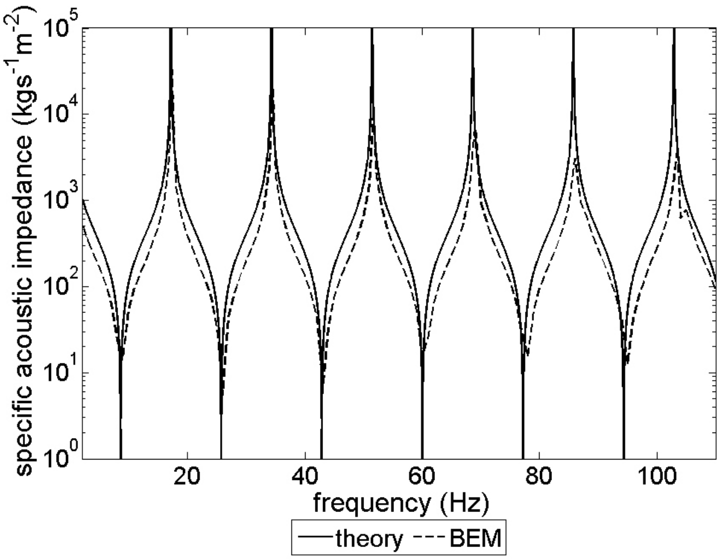

Figure 2.13: Harmonic response of an open-closed acoustic duct at the point of excitation.

Try comparing your BEM results with the theoretical solution using your preferred graphing package.

2.1 Extension

An important point to consider in BEM problems is mesh size. In order for a BEM solution to be accurate, there must be a sufficient number of elements per wavelength. Hence at higher frequency, the mesh density must be greater. Try experimenting with your mesh density and analysis frequency to see how these affect your solution. Compare the BEM results with the theoretical result for each case. It has been proposed that for accurate BEM results to be obtained there should be a minimum of six elements per wavelength. Does this hypothesis hold true for the case of an open-closed duct?

Chapter 3

Example Two: Travelling Wave in a Duct

The second example (see Figure 1.1.*b*) is a simple model of a 1D travelling wave in a rigid walled duct.

This problem introduces the concept of impedance by the addition of absorption to the downstream end of the duct studied in the first example. The wave is fully absorbed (no reflections), resulting in a travelling wave.

Either the previous example (stand.gid) can be loaded up or a completely new model can be made.

To make changes to example one, load up 'stand.gid' and save as 'trav.gid'. To start a new model open up a new project and save it in your working folder as 'trav.gid'. Follow the same steps as outlined in the first example up to and including the step where the velocity boundary conditions are defined.

Prior to leaving the boundary condition environment (ie. after the assignment of unit velocity to mimic the piston but before selecting 'finish'), absorption needs to be added to the downstream end of the duct. To do this click on 'Impedance'. Set the real normal impedance to 415.03 (the product of the speed of sound in air and the density of air, 343 and 1.21 respectively) and leave the imaginary normal impedance as 0. Click 'assign' and then using the mouse, select the surface at z=10, the far end of the duct (see Figure 3.1). Select 'finish' to complete the assignment of boundary conditions.

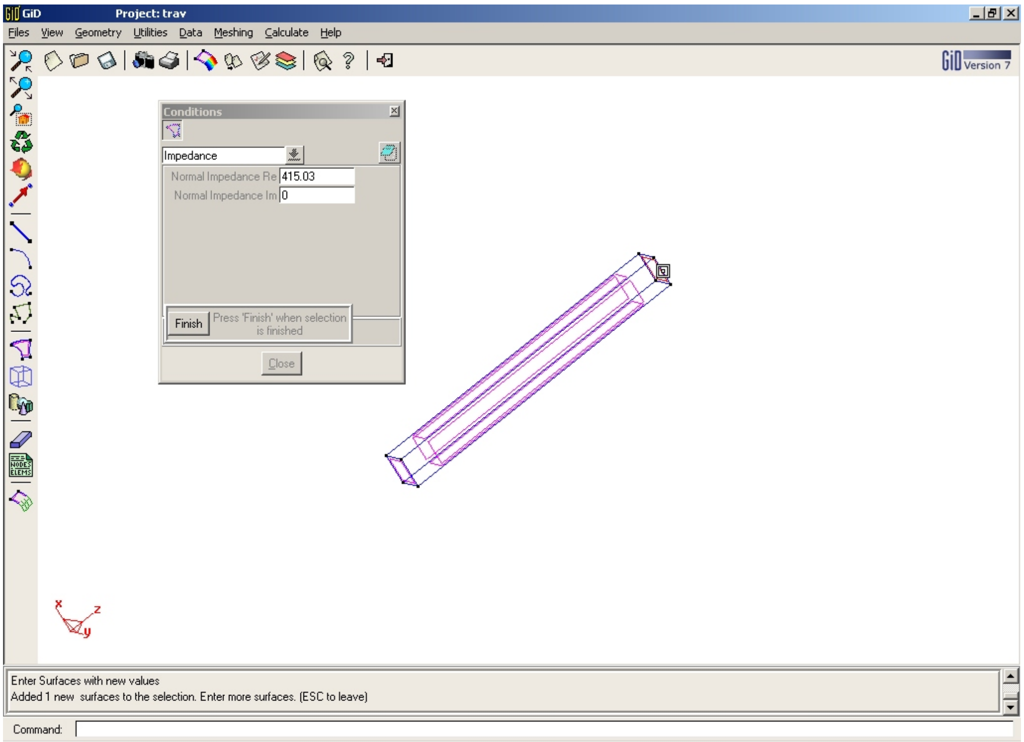

Figure 3.1: Adding an impedance to the duct.

Select (**Data > Problem data**). Give the project a title of <trav>. Set the problem data parameters to be exactly the same as for example one.

The next step is to mesh the duct. Rather than using the default triangular meshing elements, this time you'll use quadrilateral elements. To do this select **Meshing > Element type > Quadrilateral**. An information window will appear, asking you to select the surfaces to which this element type should be assigned. Click 'OK', type <:> in the command line to select all surfaces, press 'enter' and then 'esc' to leave the selection environment. Select (**Meshing > Generate**). A dialog box will appear asking you to 'Enter the size of elements to be generated'. Type in <0.5>. A dialog box will appear which states that 168 quadrilateral elements have been created. Press 'OK' and the mesh will appear.

Generate a solution and review the results in exactly the same manner as for example one. The pressure amplitude pattern obtained should differ considerably from that of the standing wave. This time, the amplitude should decrease continuously the the piston to the duct exit. (Figure 3.1)

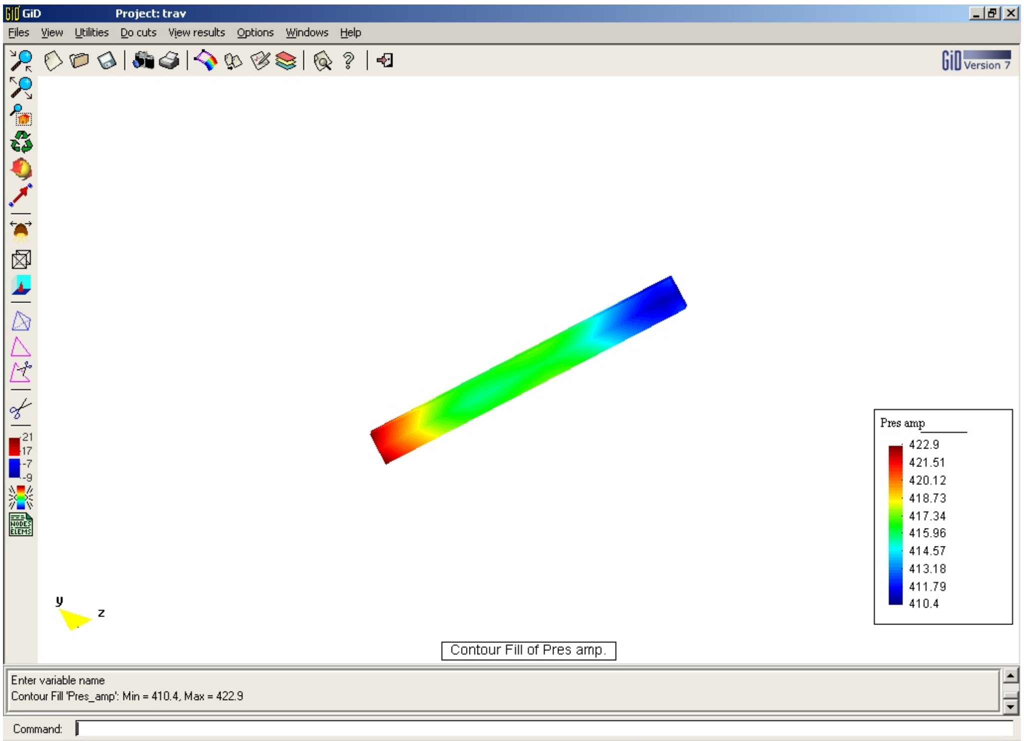

Figure 3.2: Pressure amplitude over boundary of duct.

During the generation of the solution, a file entitled 'output.dat' would have been written to your working folder. This can be opened and read using your preferred text editor. The file lists the density, speed and sound and frequency of analysis. The nodal points, element connectivity and boundary conditions are all listed within. Following this, the sound pressure on the boundary, VN on the boundary and the field point solution are listed for each frequency interval.

The analytical pressure at any point in the duct of a travelling plane wave is given by the equation:

$$p(x) = \rho c e^{-ikx} \tag{3.1}$$

where x is the distance from the point of excitation along the duct.

Using your preferred graphing package try comparing the real and complex pressures of the travelling wave (obtained by reading the BEM values of sound pressure along the centre of on of the duct sides from 'output.dat') with the analytical solution. You should obtain a graph which looks similar to Figure 3.3 (the actual sound pressure you obtain depends upon the analysis frequency chosen).

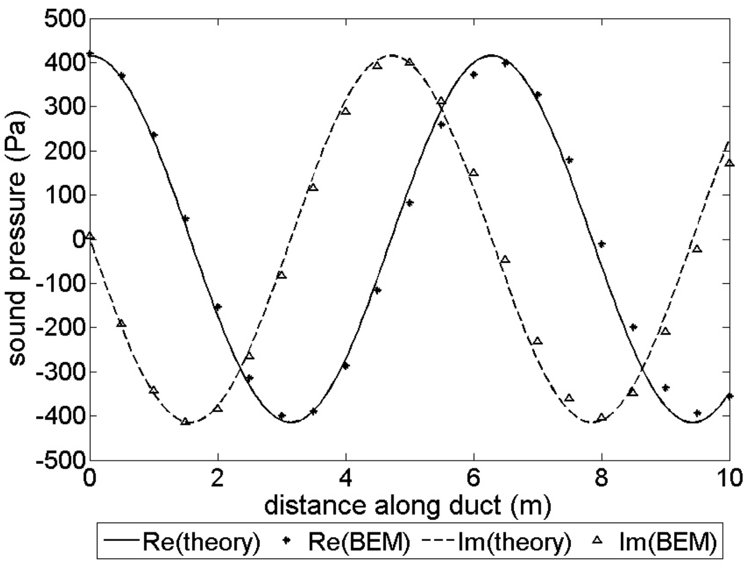

Figure 3.3: Sound pressure along the centre of duct side.

Chapter 4

Example Three: Side Branch Resonator

The third example (see Figure 1.1.c) is the addition of a side branch resonator to the travelling wave duct.

This example is currently a work in progress and will be available soon.

Chapter 5

Example Four: Speaker in a Room

The fourth example (see Figure 1.1.*d*) is a model of a speaker in the corner of a rigid walled room.

This problem introduces the excitation of modes in a 3D environment.

The first step is to construct the rectangular room. This will be done using a method different to that used to construct the rectangular tubes of examples 1 through 3, showing that there are multiple ways to construct similar geometry.

The room to be modelled is 3 metres wide, 5 metres long and 2.5 metres high. Open up a new project. Chose the option of creating a line (**Geometry > Create > Line**). Type <0,0> in the command line and then press 'enter'. A point at 0,0 will appear. Type in <2.5,0> and press 'enter'. A point at this location and a line connecting it to the previous point will appear. Type <2.5,3>, press 'enter', type <0,3> and press 'enter'. To connect the last point to the original point type <join> and press 'enter'. You will be asked to pick an existing point. Using the mouse, click on the point located at 0,0. You should now have a rectangle (see Figure 5.1). Press 'esc'

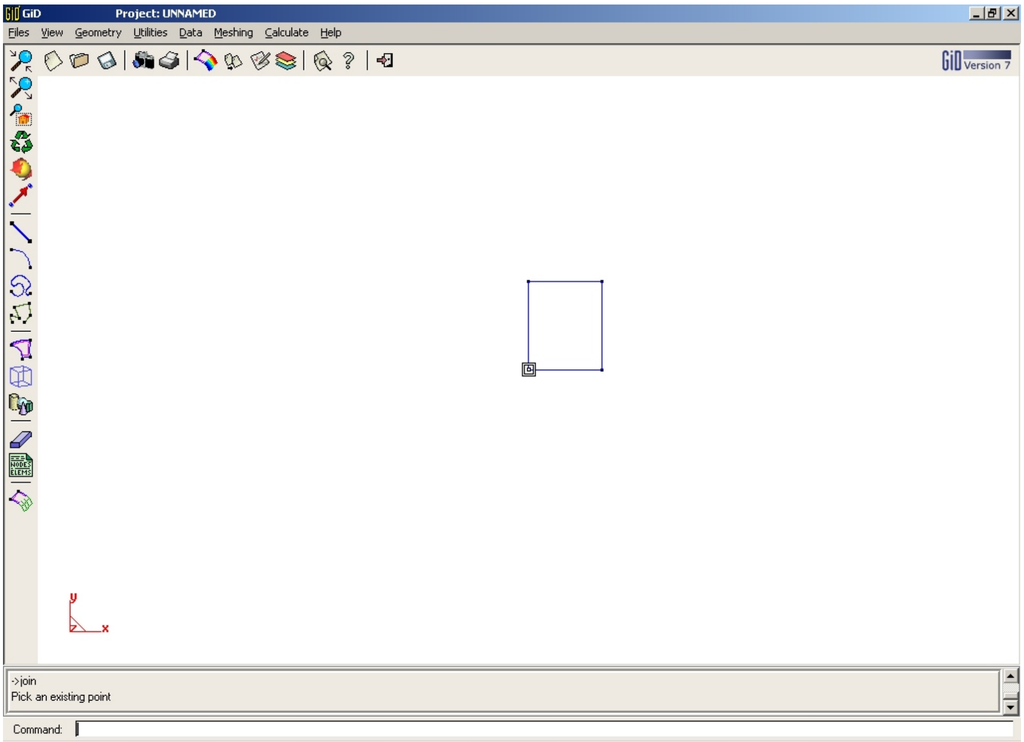

Figure 5.1: Creating the base of the room.

Save the project in your working folder as 'room.gid'.

To change to a 3D view select (**View > Rotate > Isometric**). To zoom to the best fit of the image in the window select (**View > Zoom > Frame**).

To create a surface out of the rectangle, select (**Geometry > Create > NURBS surface > By contour**). Type <:> in the command line to select all lines and press 'esc' (see Figure 5.2).

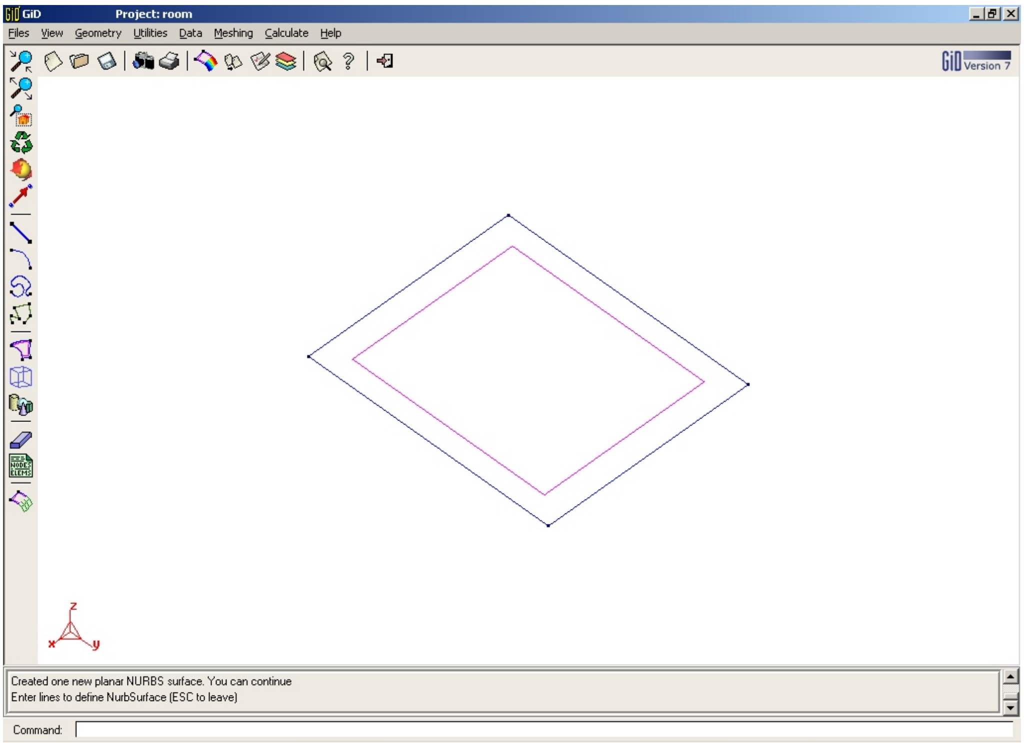

Figure 5.2: Defining the room base as a surface.

The surface can now be translated via extrusion by 5 units in the z direction. To do this type 'ctrl-c'. A copy dialog box will appear on the screen. Select the Entities type to be 'Surfaces', the Second point z coordinate to be 5.0, and Do extrude to be 'Surfaces' (see Figure 5.3). Click 'Select' and type <:> in the command line and press 'esc' to select all surfaces. Type 'Finish' and then 'Cancel'. A 3dimensional prism should have been constructed. To fit the prism in the frame select (**View > Zoom > Frame**) (see Figure 5.4).

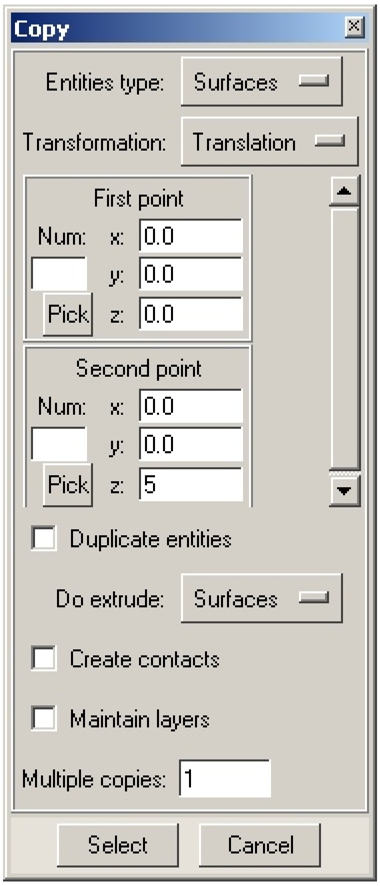

Figure 5.3: Translating the surface in the z-direction.

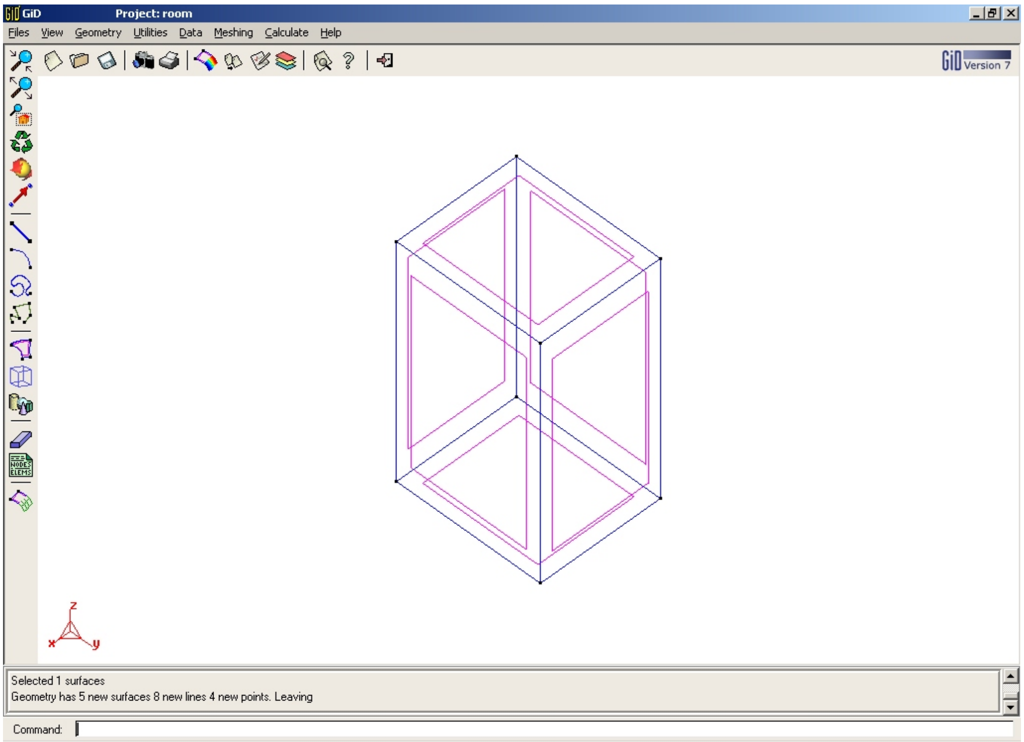

Figure 5.4: Resultant 3-dimensional room.

Rotate the view so that the x=0 plane is clearly visible using **View > Rotate > Trackball** (see Figure 5.5).

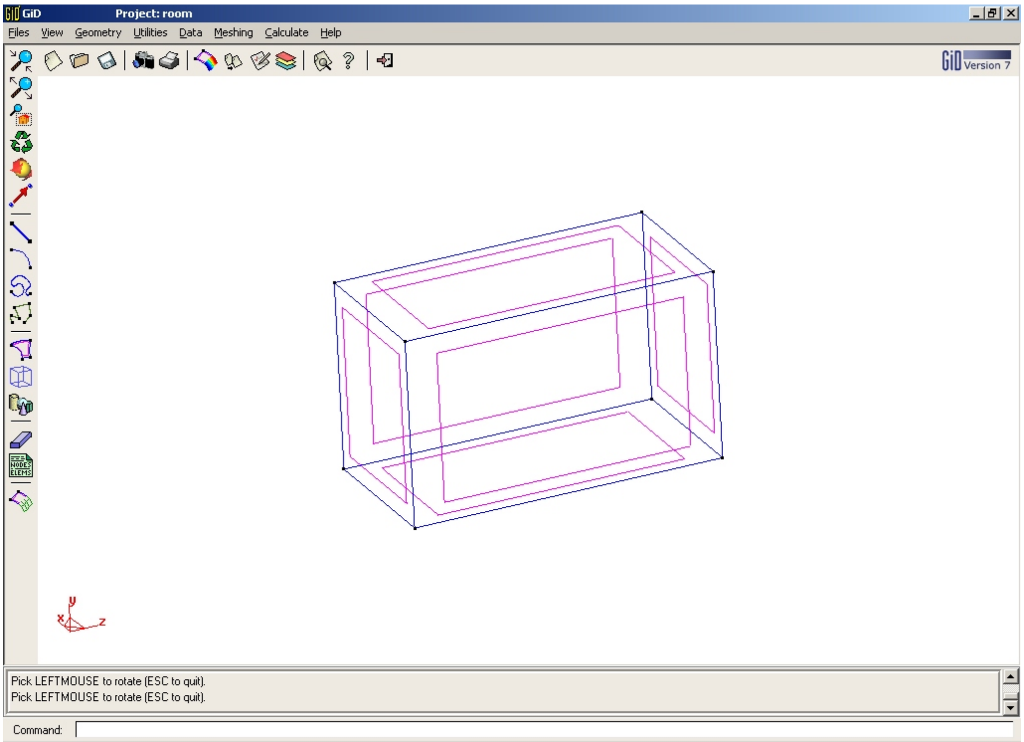

Figure 5.5: x=0 plane is clearly visible.

The next step is to define the source region. Chose the option of creating a line (**Geometry > Create > Line**). Type <0,0.1,0.1> in the command line and then press 'enter', <0,0.1,0.3>, 'enter', <0,0.3,0.3>, 'enter',and <0,0.3,0.1>, 'enter'. To connect the last point to the original point type <join> and press 'enter'. Using the mouse, click on the point located at 0,0.1,0.1. You should now have a rectangle which will be used to define the source (see Figure 5.6). Press 'esc'

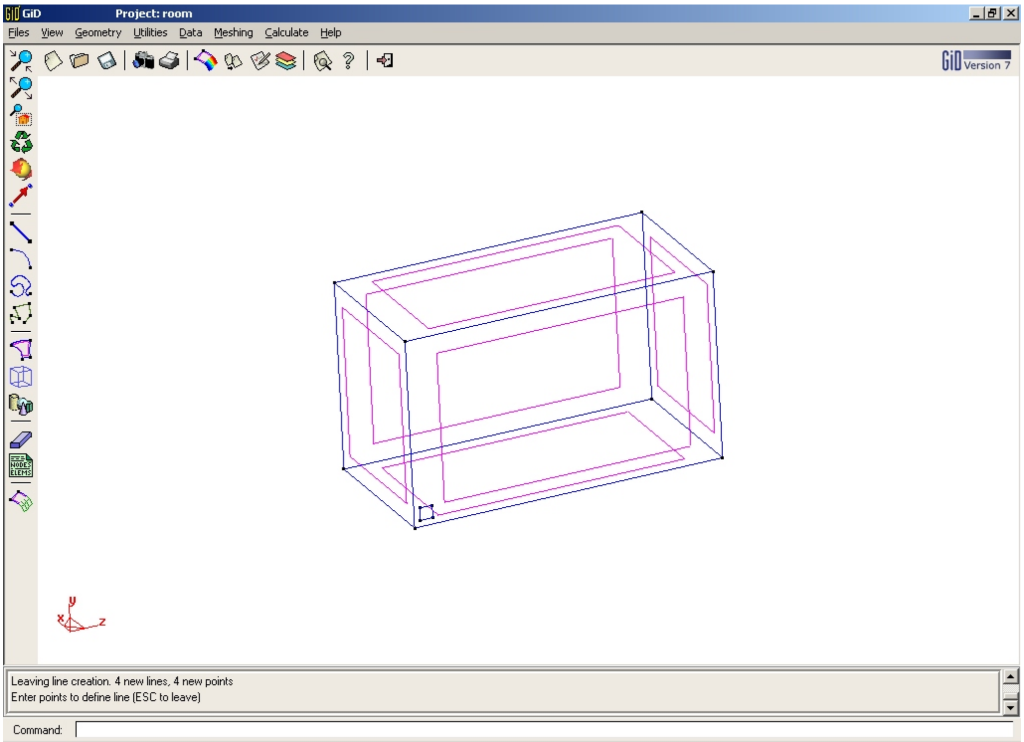

Figure 5.6: Addition of a rectangle to define the source.

The surface of the prism on which the source lies needs to be divided into two surfaces, the entire surface minus the source rectangle and the source rectangle by itself. To do this select **Geometry > Edit > Divide > Surfaces > Split**. Select the surface on the x=0 plane. Using the mouse select the four lines, defining the exterior of this surface as well as the four lines defining the source location. Press 'esc'. The surface should now have been subdivided (see Figure 5.7).

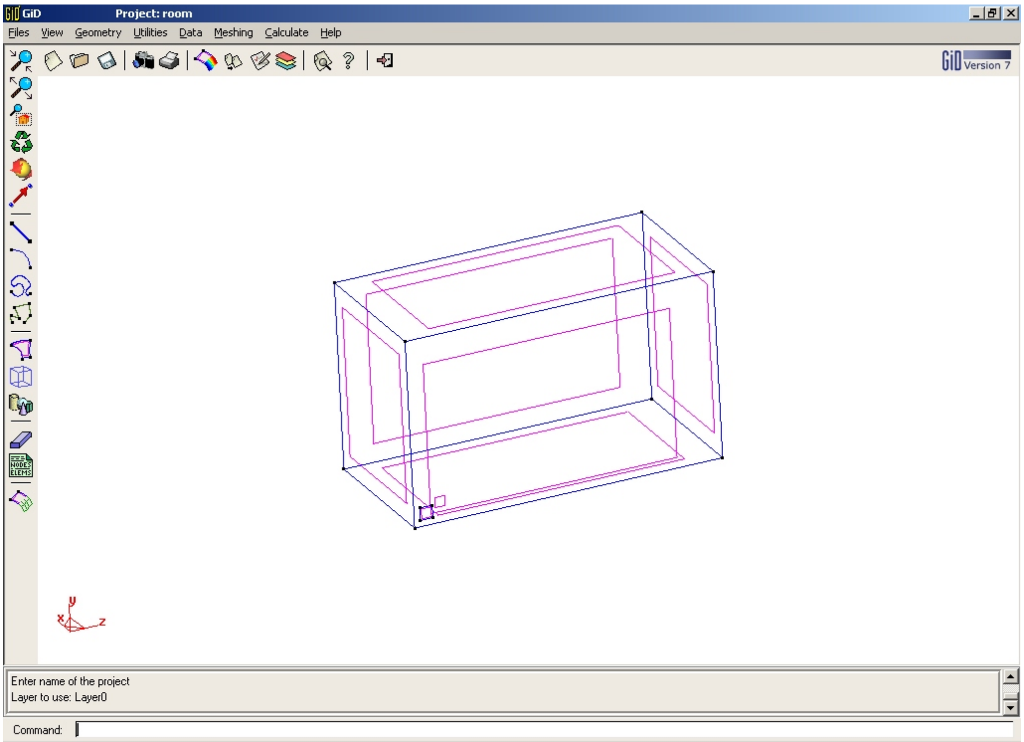

Figure 5.7: Defining the source surface.

The next step is to ensure that all of the surface normals are facing in the correct direction. Select **Utilities > Draw Normals > Surfaces**. To select all surfaces type <:> in the command line and then press 'enter'. Right click the mouse and select **Contextual > Swap some**. Click on all of the surfaces that have an inward pointing normal until all surfaces are oriented in the correct direction (see Figure 5.8). Hit 'esc' to leave the 'Draw Normals' mode.

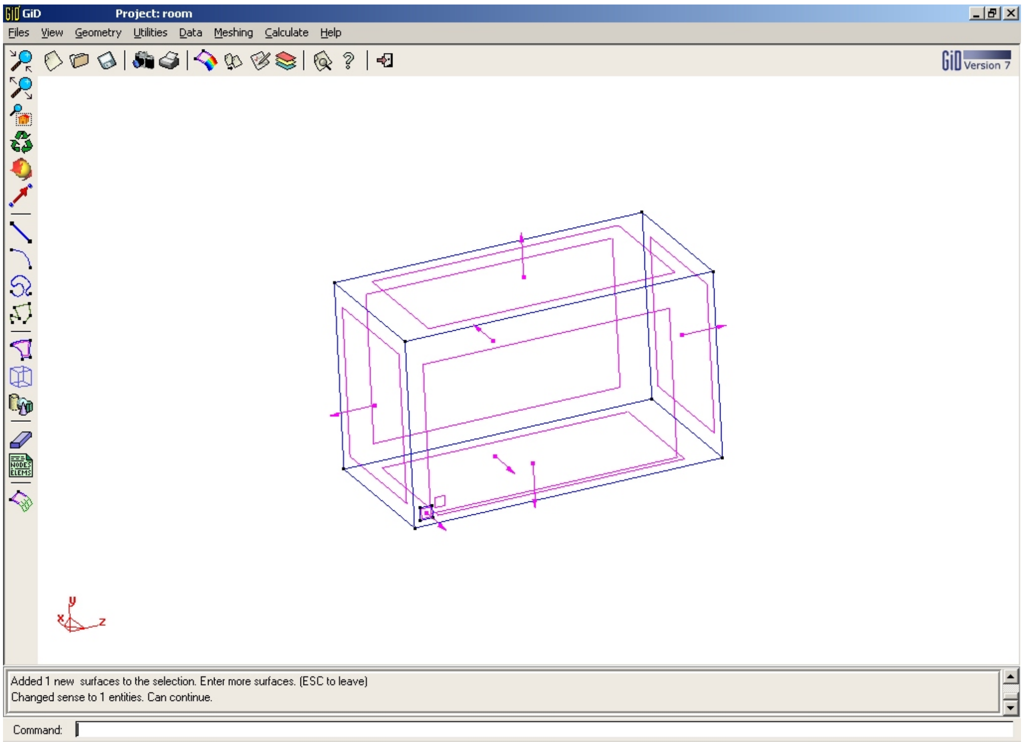

Figure 5.8: Correct orientation of the surface normals.

The next step is to define the problem as a Helm3D BEM problem. To do this select **Data > Problem type > helm3d**. A dialog window warning the user that 'all data information will be lost' will appear. Select 'OK'.

The only boundary conditions required for this problem are velocity boundary conditions. A velocity of 1+0i is needed to mimic the piston at the source location. All other surfaces need to be set to zero velocity. To do this select (**Data > Conditions**). Click on 'Velocity'. Set both the real and imaginary velocity components to zero. Click on 'assign'. Type <:> in the command line to select all surfaces and select 'finish'. Change the real velocity component to 1. Click on assign and select only the source surface at x=0 (see Figure 5.9). Select 'finish' and then 'Close'. The boundary conditions have now been set.

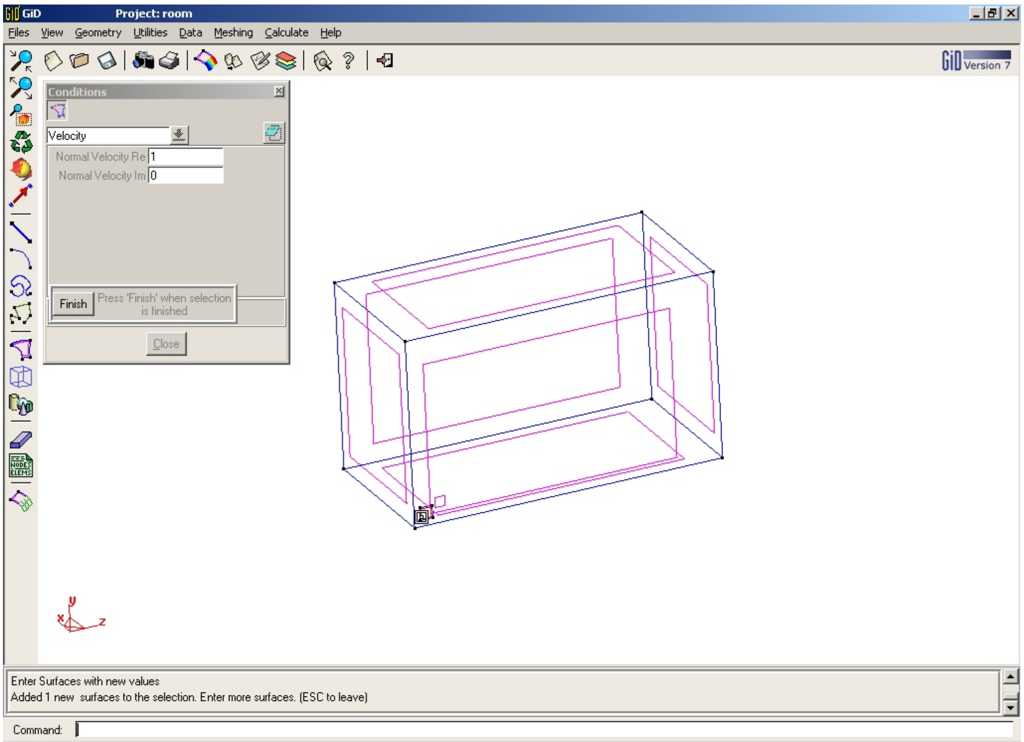

Figure 5.9: Assigning a velocity boundary condition to the source surface.

Select (**Data > Problem data**). Give the project a title of <room>. Ensure the boundary type is selected to be 'internal' and that the density and speed of sound are selected to be 1.21 and 343 respectively. Set the 'frequency start' and 'frequency stop' to be 68.5, corresponding to a wavelength of 5 m (the longest room dimension) and then click 'Accept data' and 'Close' (see Figure 5.10).

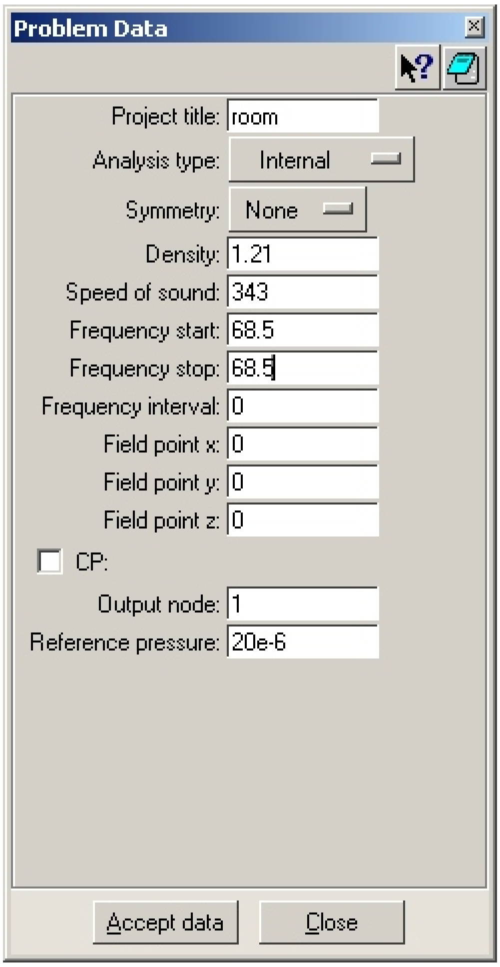

Figure 5.10: Defining the problem data.

The next step is to mesh the boundary of the room. Rather than using the default triangular meshing elements, this time you'll use quadrilateral elements. To do this select **Meshing > Element type > Quadrilateral**. An information window will appear, asking you to select the surfaces to which this element type should be assigned. Click 'OK', type <:> in the command line to select all surfaces, press 'enter' and then 'esc' to leave the selection environment. Select (**Meshing > Generate**). A dialog box will appear asking you to 'Enter the size of elements to be generated'. Type in <0.4>. A dialog box will appear which states that 498 quadrilateral elements have been created. Press 'OK' and the mesh will appear (Figure 5.11).

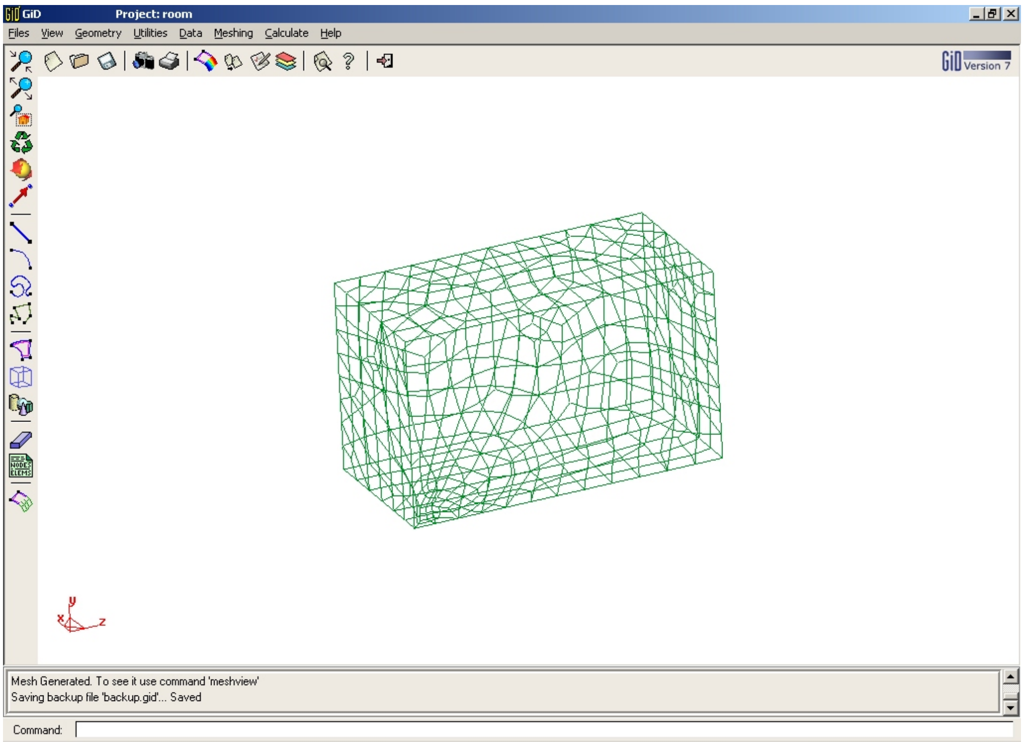

Figure 5.11: Meshing the boundary of the room.

A solution to the problem can now be generated by selecting (**Calculate > Calculate**). A dialog box will appear telling you once the solution is done. Click 'OK'. To review the results, once must enter the postprocessor. Select (**Files > Postprocess**). To display the pressure over the boundary of the room select (**View results > Contour fill > Press amp**) (Figure 5.12).

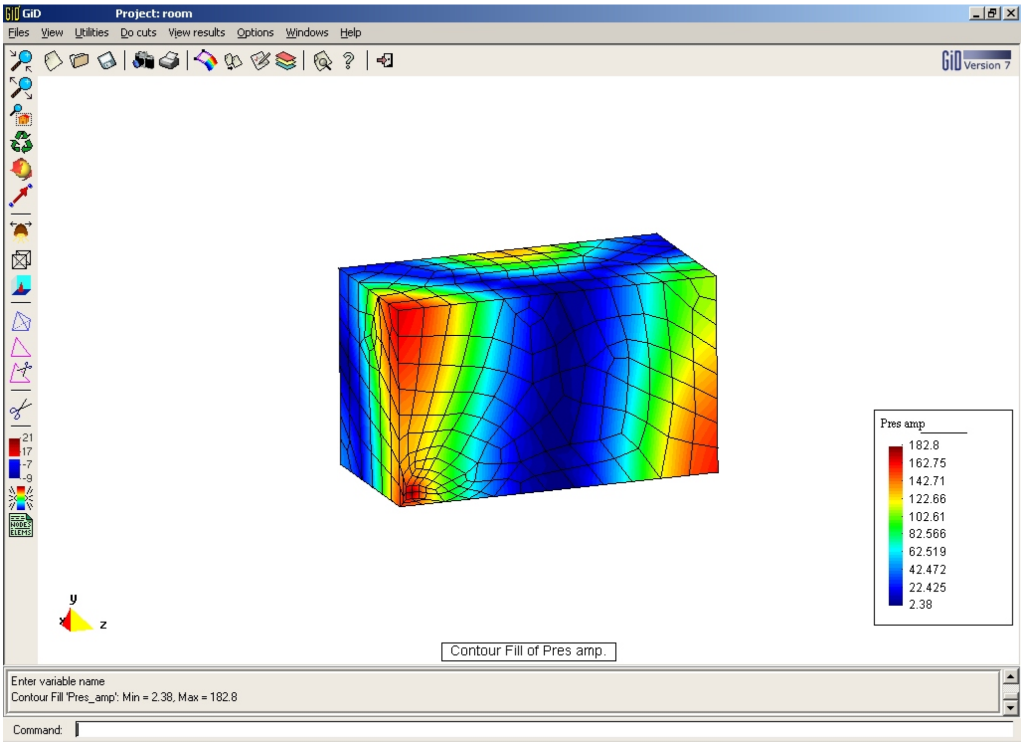

Figure 5.12: Pressure amplitude over the boundary of the room.

5.1 Extension

The solution was obtained at the resonance frequency associated with one of the room dimensions. Try comparing solutions obtained both at and away from the various resonances associated with the room. Can you see a pattern? (hint: The resonance frequencies of the room are given by the equation:

$$f_n = \frac{c}{2}\sqrt{(\frac{n_x}{L_x})^2 + (\frac{n_y}{L_y})^2 + (\frac{n_z}{L_z})^2} \tag{5.1}$$

where c is the speed of sound in the media, n_x, n_y and n_z are the (integer) mode numbers in the x, y and z axial directions respectively and L_x, L_y and L_z are the lengths of the room in the x, y and z directions respectively. The axial resonances occur when two of the mode numbers are set to zero and the other is not. Tangential resonances occur when one of the mode numbers is zero and oblique resonances occur when all mode numbers are non-zero).

Try comparing sources of identical volume velocity but different shapes (such as circular or rectangular sources) and sizes (hint: the original source is 0.2 m by 0.2 m or 0.04 m^2 in area and has a velocity of 1 m/s, corresponding to a volume velocity of 0.04 m^3/s. Hence, if you increase or decrease the total area of the source, the velocity boundary condition of the source must be decreased or increased by an amount which will maintain the 0.04 m^3/s volume velocity). Make sure that the centroid of the source location remains at approximately the same location within the room. How does changing the source geometry affect the results you obtain? What would happen if you changed the volume velocity?

The room that you analysed in this problem had three different axial dimensions. What would happen if this were not the case? Try modelling rooms with two or even three of the room dimensions being identical. See how this affects the sound pressure over the boundary both at resonance (by selecting a frequency corresponding to the ratio between the speed of sound and the repeated dimension $f = c/\lambda$) and off resonance. What can you conclude from this? If you were to design a room do you think it would be a good idea to use identical dimensions along each axis or should you purposely use unequal dimensions. What would be the effect of having a sloped roof or alcove?

Chapter 6

Example Five: Sound in a Car

The final interior problem, the interior of a car (see Figure 1.1.e) gives an example of how BEM can be applied to a practical 3D problem.

The geometry of this problem is more complicated than that of the previous problems and as such, would be cumbersome to construct within the GiD environment. More complex shapes should be drawn using another drawing package and then imported into GiD.

Open up a new project (**Files > New**) and save it (**Files > Save**) in your working directory as 'car.gid'. Import the car geometry (car.igs) into GiD (**Files > Import > IGES**). An information window specifying the read time and geometry information of the model will appear. Click 'close'. The car geometry should have appeared (see Figure 6.1).

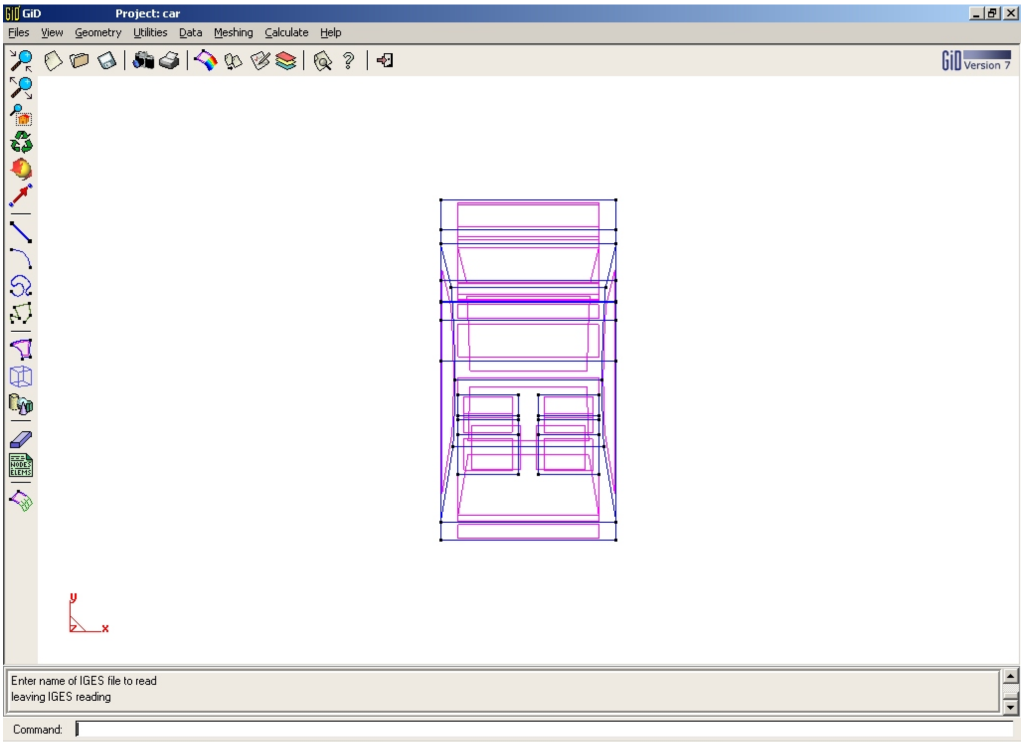

Figure 6.1: Imported car geometry.

The geometry was originally constructed in mm and hence it is necessary to scale it to metres. To see the current point coordinates select **Utilities > List > Points**, type <:> in the command line to select all points and press 'esc'. A 'list entities' window will appear, listing the coordinates of each point (see Figure 6.2)

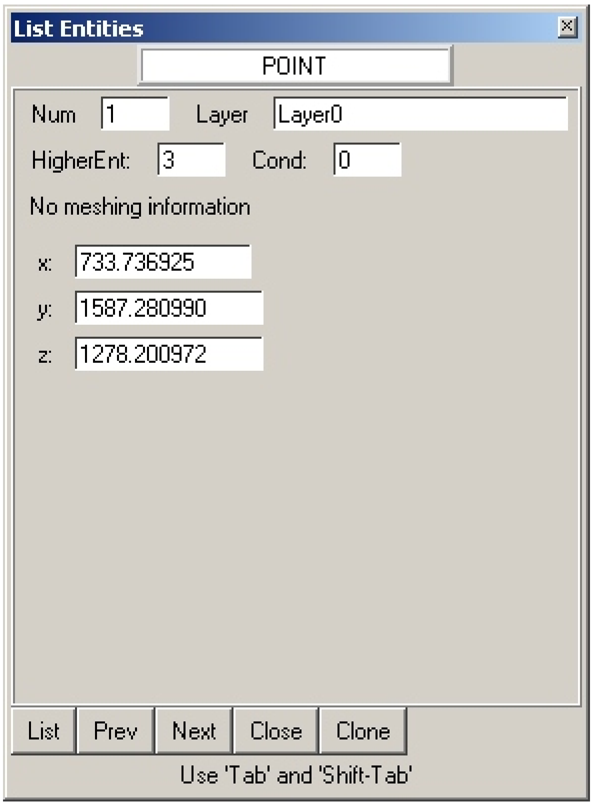

Figure 6.2: Listing the point coordinates.

To list all of the coordinates in one window (see Figure 6.3) select 'List'. Alternatively you can track backwards and forwards through each node by selecting 'Prev' or 'Next'. To close the lists select 'Close'.

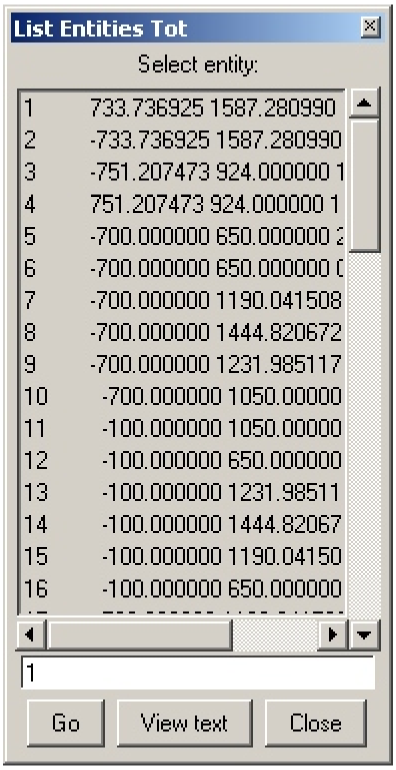

Figure 6.3: Listing the point coordinates in one window.

Select **Utilities > Move**. A window will appear. Within this window select 'Entities type' to be 'Surfaces', 'Transformation' to be 'Scale' and the 'Scale factors' to be 0.001 in the x, y and z directions. Click 'Select' (see Figure 6.4) and select the entire model by typing <:> in the command line and pressing 'enter'. Select 'finish' for the transformation to occur and then close the 'move' window by clicking the cross in the top right hand corner.

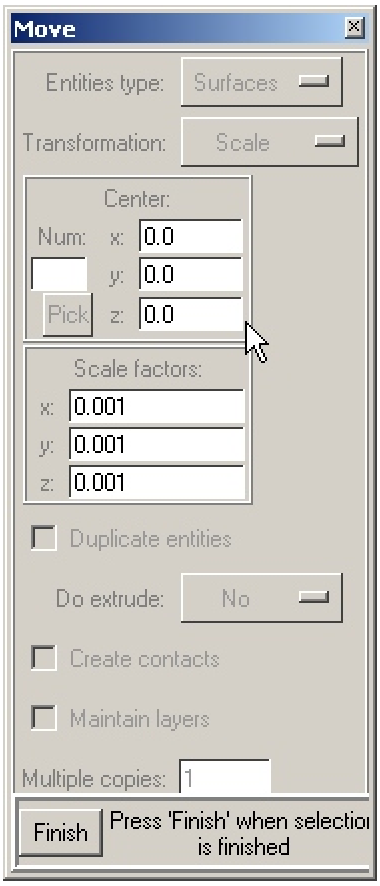

Figure 6.4: Scaling the model.

It may appear as though the car has disappeared, but this is only because it is 1/1000th of its original size. To rescale the image select **View > Zoom > Frame**. An image of the car, similar to that in Figure 6.1 should appear. To check that the scaling has been correctly implemented you can once more list the nodes by selecting **Utilities > List > Points** and 'List' (see Figure 6.5).

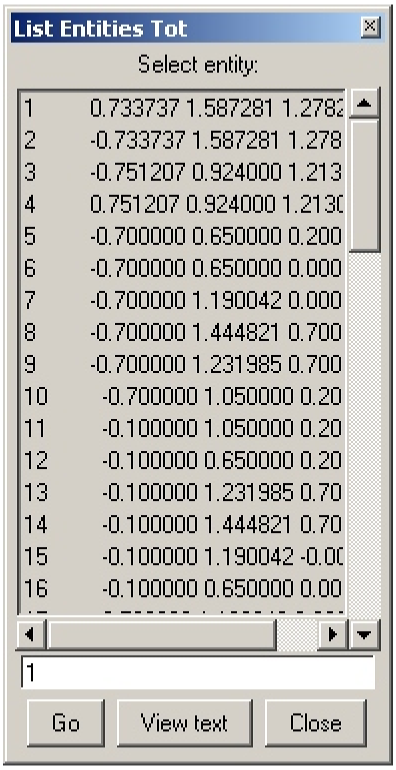

Figure 6.5: Checking the point coordinates after rescaling.

The next step is to ensure that all of the surface normals are facing in the correct direction. Select **Utilities > Draw Normals > Surfaces**. To select all surfaces type <:> in the command line and then press 'enter'. Due to there being 36 separate surfaces it is very difficult to see which surface normals are directed correctly. To rectify this problem right click the mouse, select 'Contextual' and then 'Color'. All of the outward pointing surfaces will be coloured green and all of those pointing inwards will be coloured yellow. To check all the surfaces you will need to rotate the object **View > Rotate > Trackball**. All surfaces should be orientated correctly, but if they are not (see Figure 6.6) you will need to right click the mouse and select **Contextual > Swap some**. Click on all of the yellow surfaces once to reorientate them in the correct direction (see Figure 6.7). Press 'esc' to leave the 'draw normals' environment.

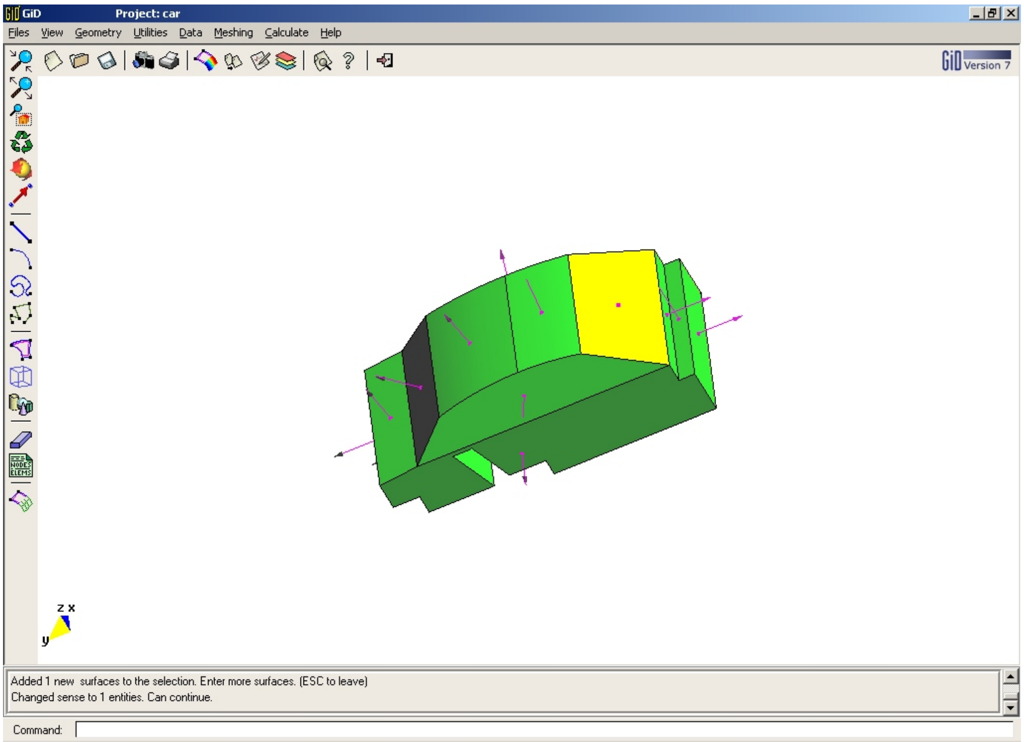

Figure 6.6: Checking the surface normals.

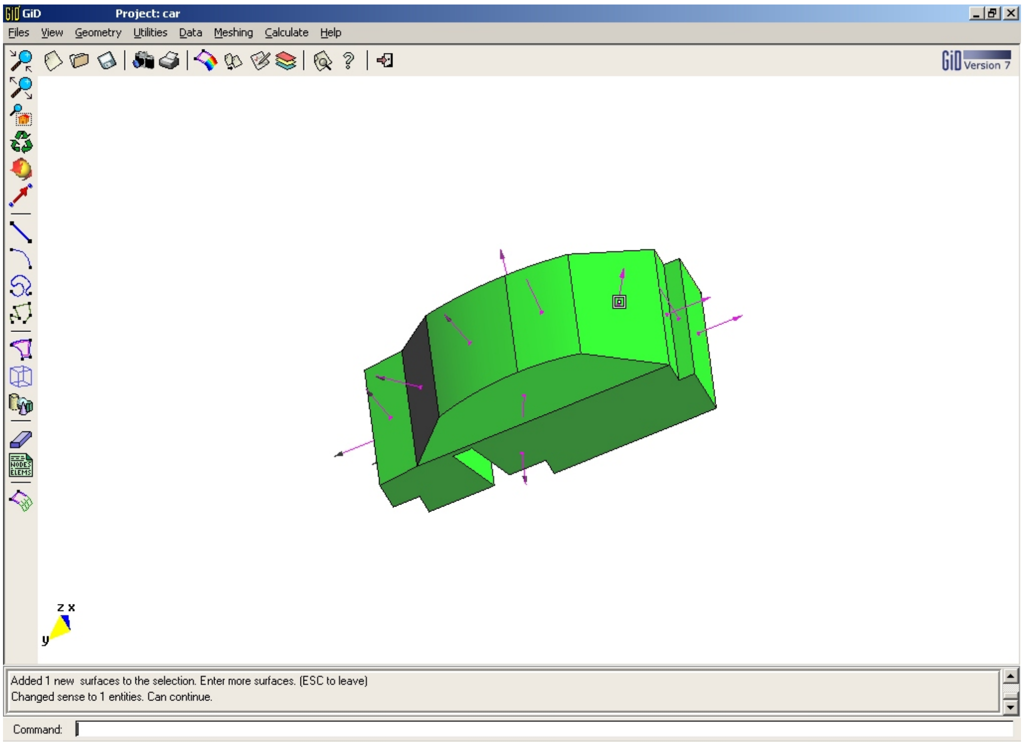

Figure 6.7: All surface normals are oriented correctly.

The next step is to define the problem as a Helm3D BEM problem. To do this select **Data > Problem type > helm3d**. A dialog window warning the user that 'all data information will be lost' will appear. Select 'OK'.

The only boundary conditions required for this problem are velocity boundary conditions. A velocity of 1+0i is used to represent sound transmission through the engine firewall. All other surfaces need to be set to zero velocity. To do this select (**Data > Conditions**). Click on 'Velocity'. Set both the real and imaginary velocity components to zero. Click on 'assign'. Type <:> in the command line to select all surfaces and select 'finish'. Change the real velocity component to 1. Click on assign and select only the vertical surface at the far rear end of the car (see Figure 6.8). Select 'finish' and then 'Close'. The boundary conditions have now been set.

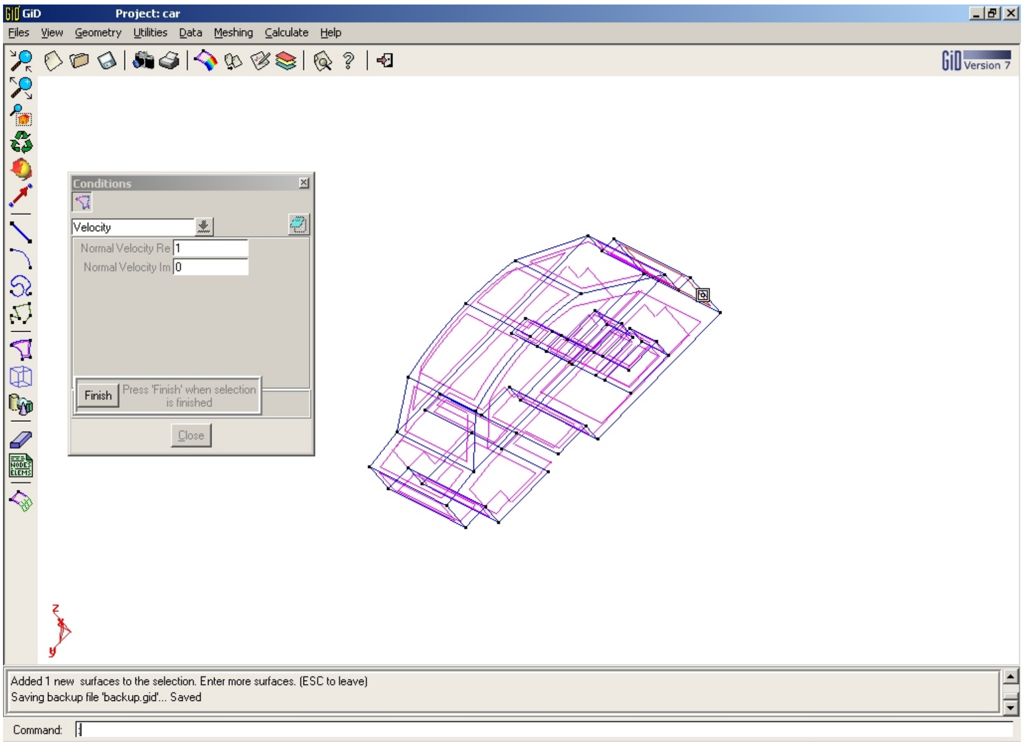

Figure 6.8: Implementing the velocity boundary condition.

Select (**Data > Problem data**). Give the project a title of <car>. Ensure the boundary type is selected to be 'internal' and that the density and speed of sound are selected to be 1.21 and 343 respectively. Set the 'frequency start' and 'frequency stop' to be 100 and then click 'Accept data' and 'Close' (see Figure 6.9).

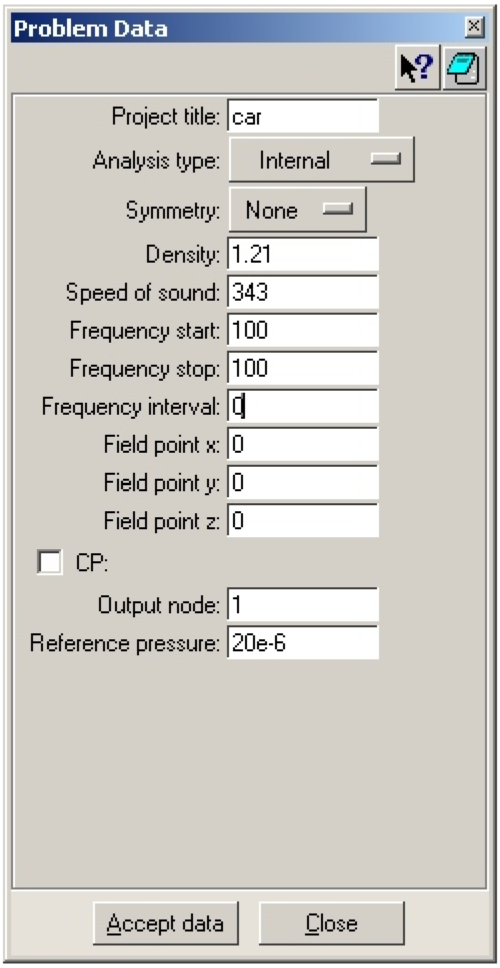

Figure 6.9: Problem data for car.

The next step is to mesh the surface of the car. Select (**Meshing > Generate**). A dialog box will appear asking you to 'Enter the size of elements to be generated'. Type in <0.4> and press 'OK'. A dialog box will appear which states that 490 triangle elements have been created. Press 'OK' and the mesh will appear (Figure 6.10).

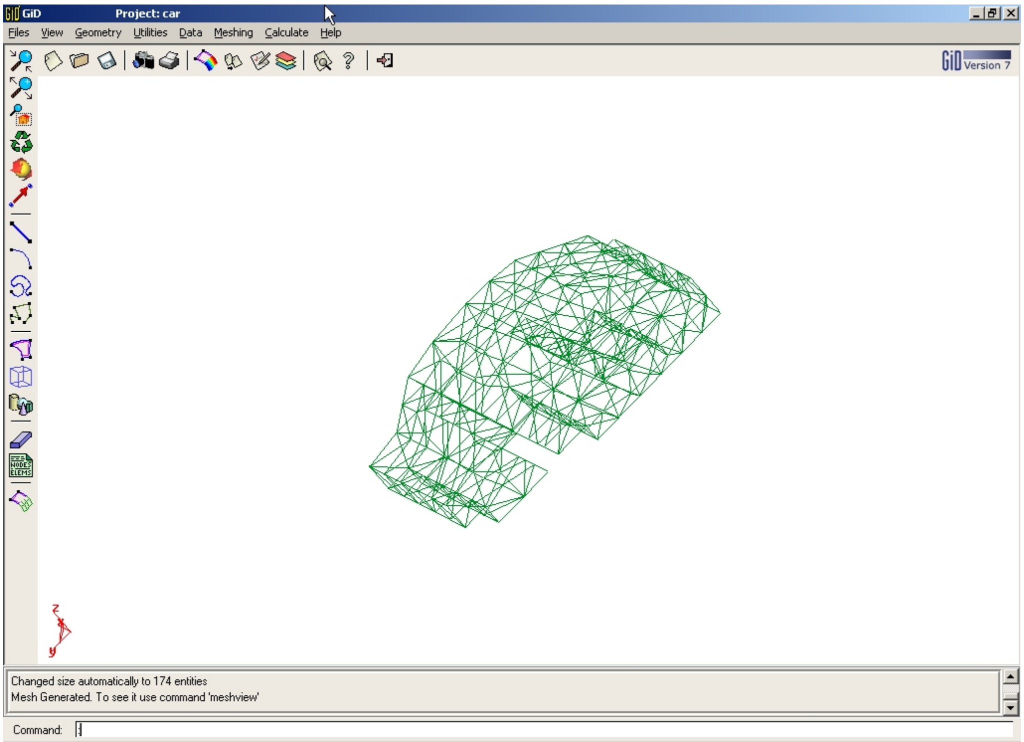

Figure 6.10: Meshed car boundary.

A solution to the problem can now be generated by selecting (**Calculate** > **Calculate**). A dialog box will appear telling you once the solution is done. Click 'OK'. To review the results, once must enter the postprocessor. Select (**Files** > **Postprocess**). To display the pressure over the boundary of the car select (**View results** > **Contour fill** > **Press amp**) (Figure 6.11).

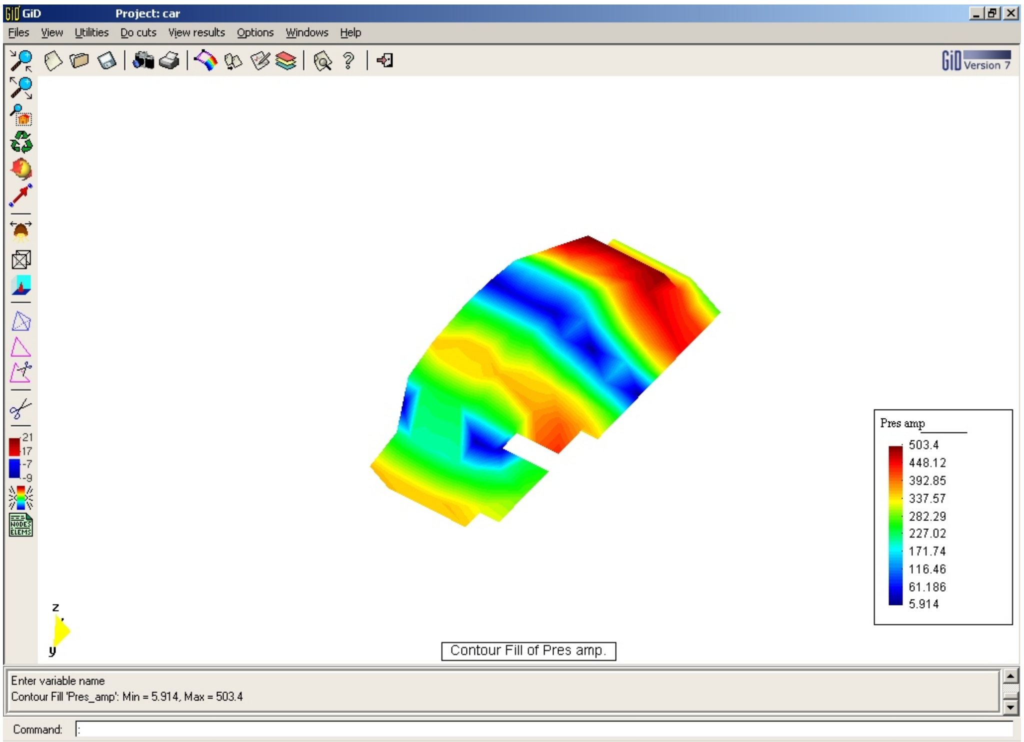

Figure 6.11: Pressure magnitude on car interior.

6.1 Extension

The pressure over the car interior is highly frequency dependent. Try solving the BEM problem at other frequencies and compare the pressure plot to that obtained at 100Hz. To do this, leave the preprocessor, redefine the 'problem data' frequency to that which you desire, remesh and then resolve the problem. Note how at certain frequencies, very high pressure amplitudes are obtained at various locations across the car interior boundary. How could this present itself as a problem in a real situation and what feasible solutions could you implement to ameliorate this problem?

Chapter 7

Example Six: Pulsating Sphere

The first exterior problem is the classical fundamental radiation problem of a pulsating sphere (see Figure 1.1.f). Key concepts covered are modelling symmetry and how this affects computational efficiency, appropriate direction of normals for an external problem and the use of CHIEF points in the interior to improve the condition number of the matrix.

To model a complete sphere select **Geometry > Create > Object > Sphere**. You will be asked to enter a centre for the sphere. In the command line type <0,0,0> and press 'enter'. You will then be asked to enter a radius for the sphere. Type <1> and press 'enter'. A sphere should appear on the screen. To enlarge the view select **View > Zoom > Frame** (see Figure 7.1).

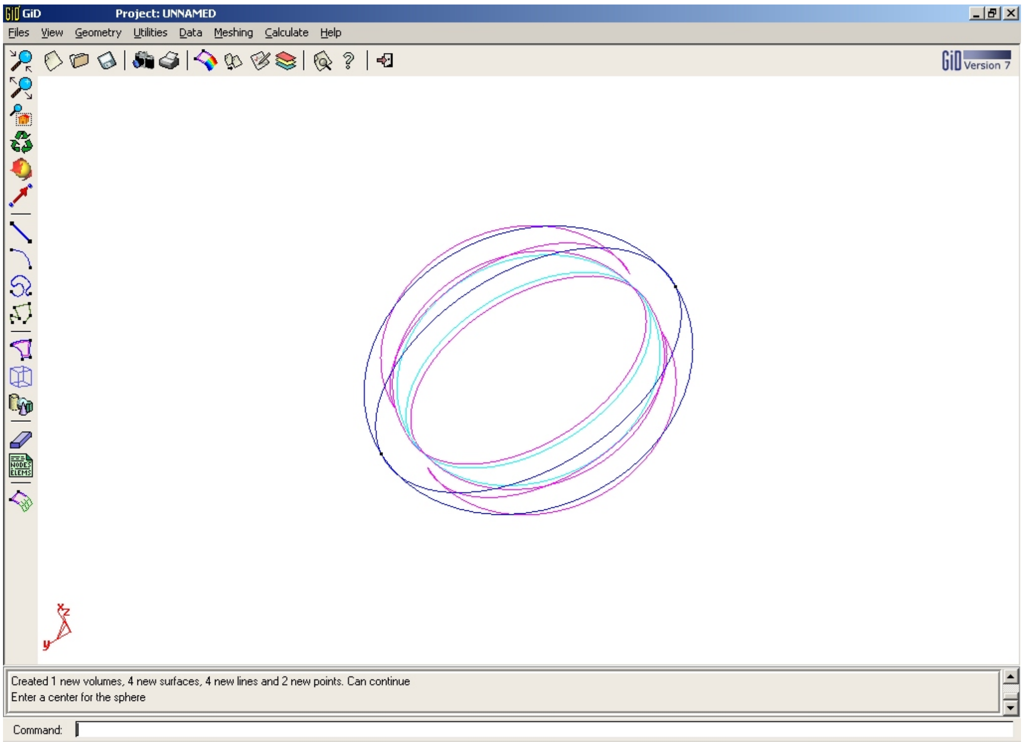

Figure 7.1: Generated sphere.

Since the sphere is symmetrical, it can be modelled using half a sphere with a symmetry boundary condition. Half the sphere must therefore be deleted. Select **Geometry > Delete > Volume**, click on the blue lines defining the spherical volume and press 'esc'. Select **Geometry > Delete > Surface**, click on the two surfaces in the negative x-coordinate region (see Figure 7.2) and press 'esc'.

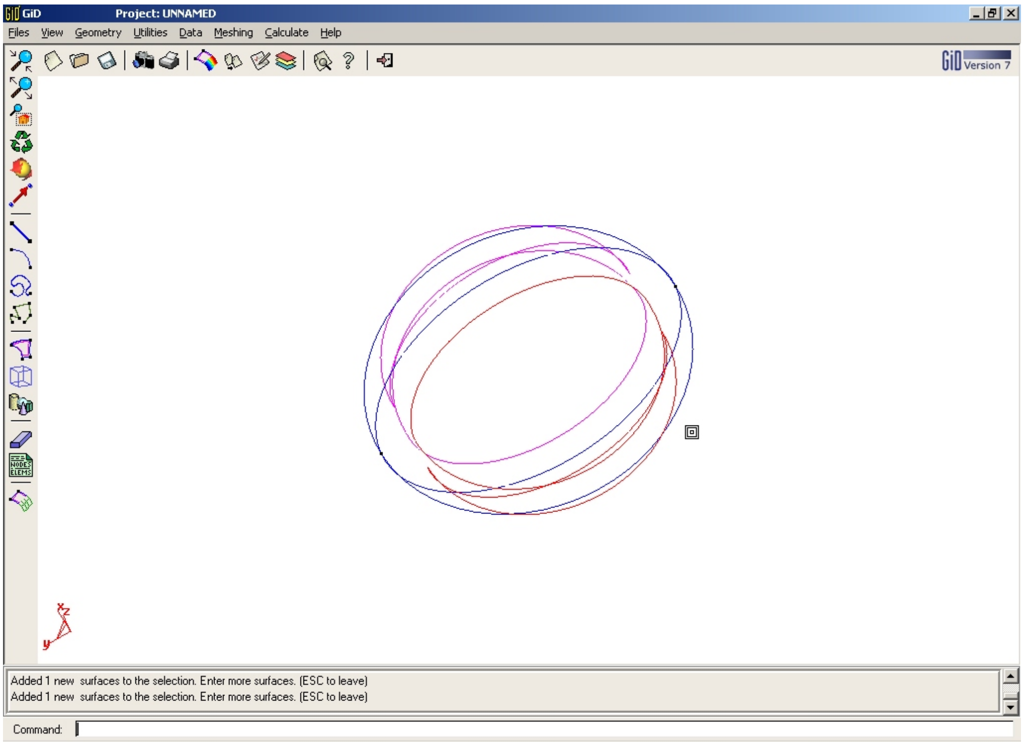

Figure 7.2: Deleting sphere surfaces in the negative x-coordinate region.

Select **Geometry** > **Delete** > **Line**, click on the line in the negative x-coordinate region and press 'esc'. You should be left with a semi-sphere as depicted in Figure 7.3

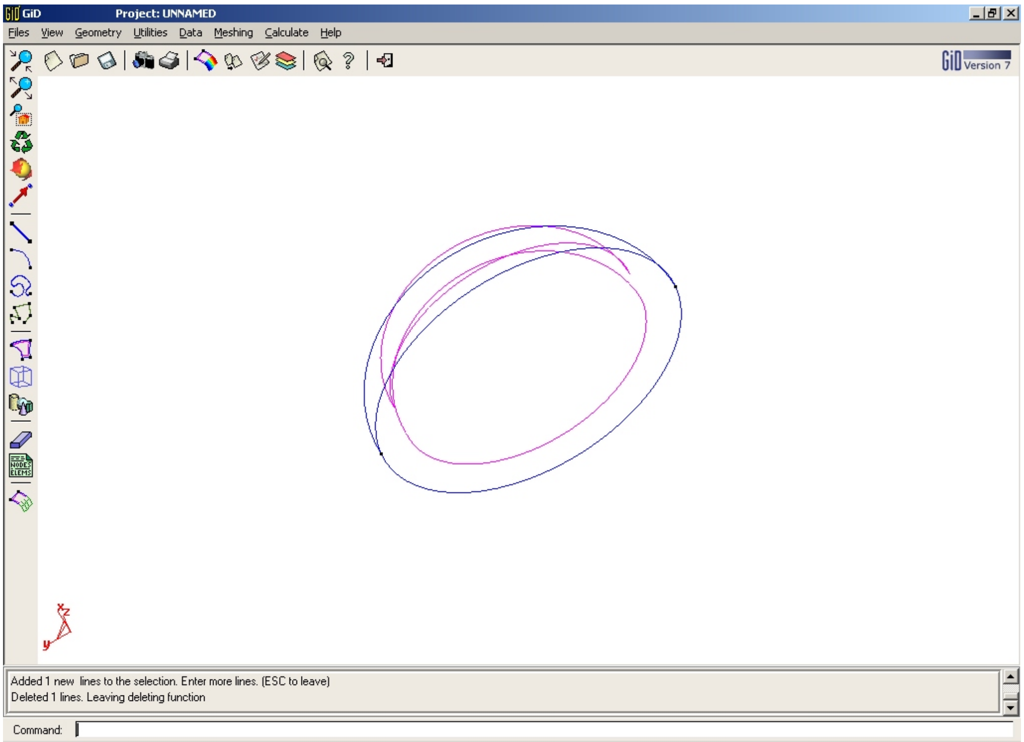

Figure 7.3: Model of half a sphere.

Save the project in your working folder as 'sphere.gid'.

The next step is to check the surface normals. For an external boundary element problem all surfaces must face in. Select **Utilities > Draw Normals > Surfaces**. To select all surfaces type <:> and then press 'enter'. To see the directions of both surface normals clearly you may need to change the rotation of the sphere. To do this select **View > Rotate > Trackball** and rotate the view until you are happy that the normals can be clearly seen. Right click the mouse and select **Contextual > Swap some**. Click on all of the surfaces that have an outward pointing normal until all surfaces are oriented in the correct direction (see Figure 7.4). Hit 'esc' to leave the 'Draw Normals' mode.

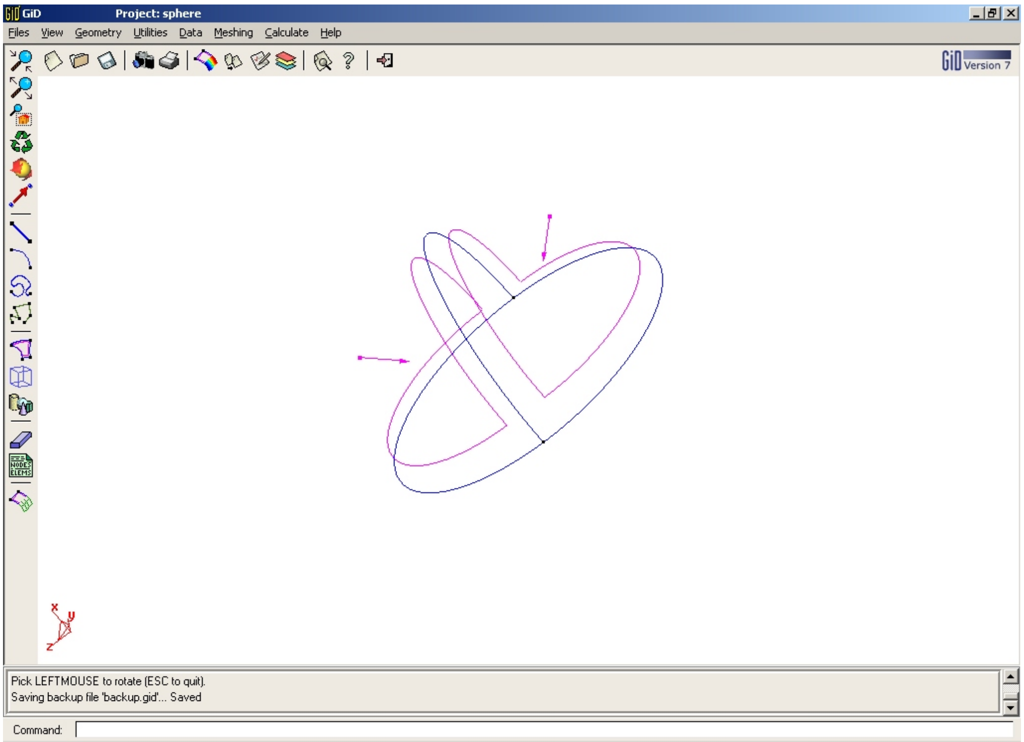

Figure 7.4: Orienting the surface normals in the correct direction.

The next step is to define the problem as a Helm3D BEM problem. To do this select **Data > Problem type > helm3d**. A dialog window warning the user that 'all data information will be lost' will appear. Select 'OK'.

The only boundary conditions required for this problem are velocity boundary conditions. A velocity of 1+0i is needed over the entire surface of the sphere. To do this select (**Data > Conditions**). Click on 'Velocity'. Set the real velocity component to 1 and the imaginary velocity components to zero. Click on 'assign'. Type <:> in the command line to select all surfaces and select 'finish' and then 'Close'. The boundary conditions have now been set.

Select (**Data > Problem data**). Give the project a title of <sphere>. Ensure the boundary type is selected to be 'external', symmetry is set to 'Y-Z', density is set to 1, speed of sound is set to 6.2832, freq start to 0.1, freq stop to 7 and freq int to 0.1 (see Figure 7.5). Click 'Accept data' and 'Close' to leave the problem data environment.

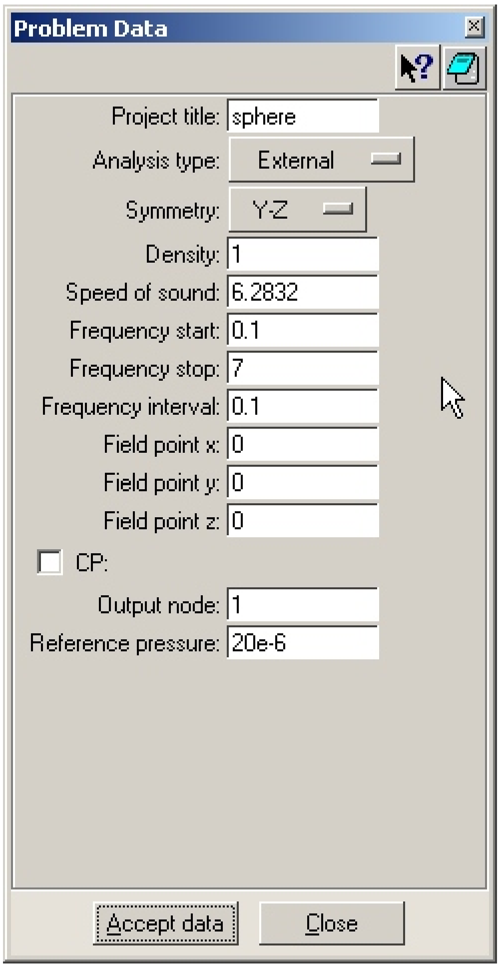

Figure 7.5: Selecting the problem data.

The next step is to mesh the semi-sphere surface. Select (**Meshing > Generate**). A dialog box will appear asking you to 'Enter the size of elements to be generated'. Type in <0.2> and press 'OK'. A dialog box will appear which states that 568 triangle elements have been created. Press 'OK' and the mesh will appear (Figure 7.6).

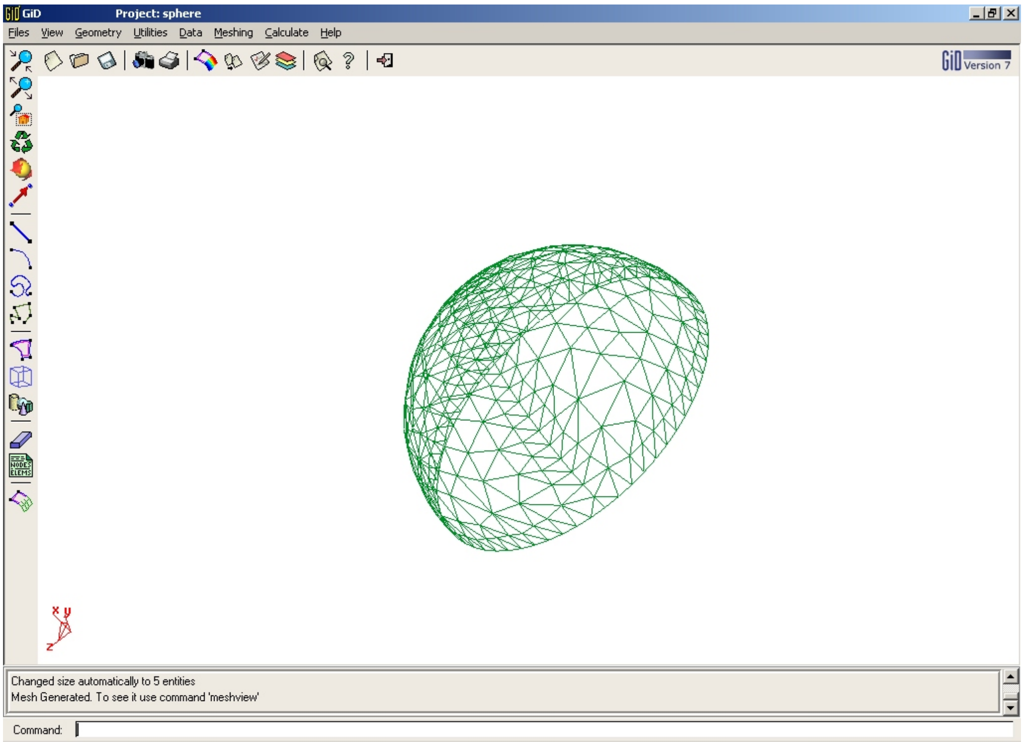

Figure 7.6: Meshed semi-sphere.

A solution to the problem can now be generated by selecting (**Calculate > Calculate**). A dialog box will appear telling you once the solution is done. Click 'OK'.

The file 'sphere.fdat' will have been created in your working directory. You can open this using a text editor to review your results. Numerous columns of data appear, each labelled with a heading in the first row. Each row of data corresponds to one calculation frequency. Of particular interest are the first, second and ninth columns, the frequency, condition number and pressure amplitude respectively. The condition number is important as it gives an indication of how well conditioned the matrix to be inverted is. The closer the condition number is to unity, the better the conditioning. The higher the condition number, the more poorly conditioned the matrix is and hence the results obtained at these frequencies are more likely to be incorrect. Note that at certain frequencies the condition number is high. Save the file in a separate folder for comparative puposes which will be explained later.

One disadvantage to the direct BEM approach is that if the Kirchoff-Helmholtz integral equation is used to represent the sound field on the exterior of a finite volume, at the natural frequencies of the interior of the finite volume, the exterior problem breaks down and the matrix becomes ill-conditioned. This is well documented [9] and many solutions have been attempted [10, 11]. The CHIEF method [10] is commonly used to overcome the interior natural frequency problem because of its simplicity. This technique solves an overdetermined system of equations formed by placing extra points (CHIEF points) inside the volume of interest. Provided the CHIEF points are not placed at a nodal line of the interior solution, this will improve the matrix condition number and allow the matrix to be solved using least-squares methods.

Re-enter the preprocessor and alter the problem data to include a CHIEF point. check the 'CP' box to indicate the inclusion of a chief point, set Chief point x to be 0 and output node to be 1. Remesh and resolve the problem. Save the newly generated *.fdat file for future comparison. Repeat the process with Chief point x at 0.5.

The analytical solution for the pressure produced by a pulsating sphere, which can be derived from the spherical wave equation, is:

$$p(r) = (\frac{a^2}{r})\frac{i\rho\omega}{1+ika}e^{-ik(r-a)} \tag{7.1}$$

where a is the sphere radius, r is the radius at which the pressure is being calculated and ω is the angular frequency. The characteristic eigenfrequen-

cies of the sphere, which are the eigenfrequencies of the interior Dirichlet problem, are given by the equation:

$$sin(ka) = 0. \tag{7.2}$$

Compare the theoretical and BEM results (saved in the *.fdat files generated with no CHIEF point, a CHIEF point at the sphere centre and a CHIEF point at half the sphere radius) graphically using your preferred graphing package by graphing the pressure as a function of frequency or ka where k is the wavelength and a is the sphere radius. You should obtain a plot similar to Figure 7.7.

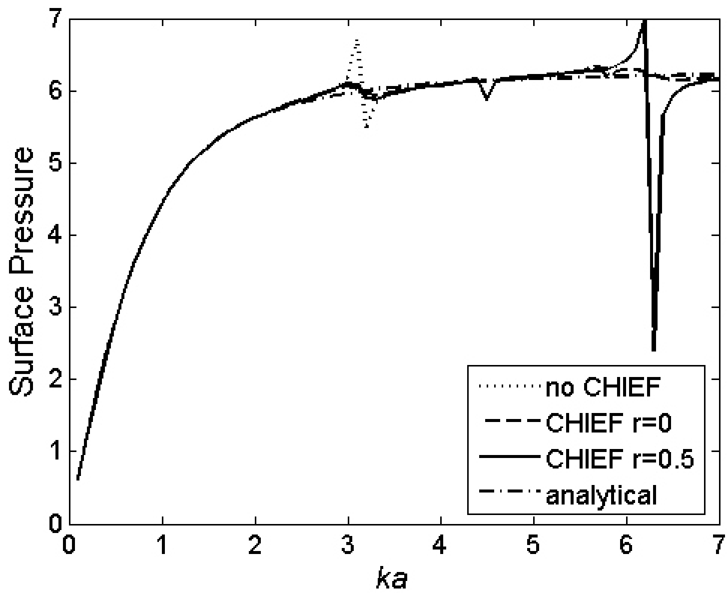

Figure 7.7: Surface pressure of a pulsating sphere.

The BEM solution with no CHIEF point shows poor agreement with the analytical solution at ka= and ka=2 , where k is the wavenumber and a is the radius of the source. This is due to poor conditioning of the matrix. The placement of a CHIEF point at r/a = 0.5, where r is the radial location from the centre of the sphere, ameliorates the problem at ka= ; however, poor agreement at ka=2 still occurs due to the CHIEF point being on the interior nodal surface corresponding to the characteristic eigenfrequency ka=2 , meaning that this resonance cannot be cancelled. Placing the CHIEF point at the sphere centre ensures that it does not lie on a nodal surface. The resulting solution is therefore in good agreement with the analytical solution. When using BEM to analyse more complex geometries, the user

generally has no prior knowledge of the optimal CHIEF point location, and therefore multiple CHIEF points randomly distributed within the volume are used.

Chapter 8

Example Seven: Model Loudspeaker

The second exterior problem is of a spherical volume with an external velocity over a proportion of its surface, representing a simplified model of a loudspeaker in a rigid walled box (see Figure 1.1.g).

This problem highlights the frequency dependance of radiation.

This example is currently a work in progress and will be available soon.

Chapter 9

Example Eight: Actual Loudspeaker

The final exterior problem applies external BEM to a more realistic situation by analysing radiation from a speaker of more realistic geometry (see Figure 1.1.h).

This example is currently a work in progress and will be available soon.

Chapter 10

Conclusion

The tutorial material described within this document covered some fundamental acoustic problems and how these would be solved using the newly developed BEM interface. The tutorials should have equipped the user with the necessary understanding and tools required for reliable application of BEM to other more complex systems.

Modelling of systems that have complexities beyond the limitations of the Helm3D BEM code will require the use of larger commercial BEM codes, however the BEM and acoustic fundamentals obtained using Helm3D will form a solid basis for any future acoustic BEM analyses, regardless of the chosen program.

References

[1] J. A. Pederson and G. Munch. Driver directivity control by sound redistibution. In *113th Convention of the Audio Engineering Society*, Los Angeles, California, October 2002. *(Cited on page 1)*

[2] T. H. Hodgson and R. L. Underwood. BEM computations of a finite length acoustic horn and comparison with experiment. In *Computational Acoustics and its Environmental Applications*, pages 213–222. WIT Press, 1997. *(Cited on page 1)*

[3] P. M. Juhl. *The boundary element method for sound field calculations.* PhD thesis, Technical University of Denmark, 1993. *(Cited on page 1)*

[4] R. D. Ciskowski and C. A. Brebbia, editors. *Boundary element methods in Acoustics.* Computational Mechanics Publications, Co-published with Elsevier Applied Science, 1991. *(Cited on page 1)*

[5] O. von Estorff, editor. *Boundary Elements in Acoustics: Advances and Applications.* WIT Press, 2000. *(Cited on page 1)*

[6] R. C. Morgans. External acoustic analysis using comet. Technical report, Internal Report, The University of Adelaide, 2000. *(Cited on page 1)*

[7] T. W. Wu, editor. *Boundary Element Acoustics: Fundamentals and Computer Codes.* WITPress, 2000. *(Cited on pages 1 and 2)*

[8] R. C. Morgans, A. C. Zander, and C. H. Hansen. Fast boundary element models for far field pressure prediction. In *Australian Acoustical Society Conference, Acoustics 2004*, 2004. *(Cited on page 1)*

[9] L. G. Copley. Fundamental results concerning integral representations in acoustic radiation. *Journal of the Acoustical Society of America*, 44 (1):28–32, July 1968. *(Cited on page 63)*

[10] H. A. Schenck. Improved integral formulation for acoustic radiation problems. *Journal of the Acoustical Society of America*, 44(1):41–58, 1968. *(Cited on page 63)*

[11] A. J. Burton and G. F. Miller. The application of integral equation methods to the numerical solutions of some exterior boundary-value problems. *Proceedings of the Royal Society of London*, A, 323:201–210, 1971. *(Cited on page 63)*